

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

До захисту допущено:

Завідувач кафедри

_____Сергій СТИПЕНКО

«__»_____2020 р.

Дипломний проєкт

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Програмне забезпечення
високопродуктивних комп'ютерних систем і мереж»**

спеціальності 121 «Інженерія програмного забезпечення»

на тему: «Система комунікацій з зворотнім зв'язком»

Виконав:

студент IV курсу, групи ІП-62

Куш Родіон Вячеславович _____

Керівник:

Старший викладач,

Виноградов Юрій Миколайович _____

Консультант з нормоконтролю:

Професор кафедри ОТ, д.т.н.,

Сімоненко Валерій Павлович _____

Рецензент:

Посада, науковий ступінь, вчене звання,

Прізвище, ім'я, по батькові _____

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма «Програмне забезпечення
високопродуктивних комп'ютерних систем і мереж»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Сергій СТИПЕНКО

«__» _____ 20__ р.

ЗАВДАННЯ
на дипломний проєкт студенту
Кущу Родіону Вячеславовичу

1. Тема проєкту «Система комунікацій з зворотнім зв'язком», керівник проєкту Виноградов Юрій Миколайович, старший викладач, затверджені наказом по університету від «07» травня 2020 р. № 1081-с

2. Термін подання студентом проєкту _____

3. Вихідні дані до проєкту: технічна документація, теоретичні дані, розроблена система комунікацій із зворотнім зв'язком.

4. Зміст пояснювальної записки: огляд предметної області, аналіз існуючих рішень, проєктування і реалізація системи комунікацій із зворотнім зв'язком, інструкція користувача

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо)

Алгоритм створення зворотного зв'язку(1 шт.)

Діаграма бази даних(1 шт.)

Структурна схема системи(1 шт.)

6. Консультанти розділів проєкту*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Сімоненко В.П., проф.		

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Затвердження теми роботи	01.09.2019	
2	Вивчення та аналіз завдання	01.01.2019	
3	Аналіз існуючих передумов для розробки	28.02.2020	
4	Проектування архітектури системи	20.03.2020	
5	Розробка додатку	25.04.2020	
6	Оформлення матеріалів роботи	22.05.2020	
7	Передзахист	26.05.2020	
8	Захист		

Студент

Родіон КУЩ

Керівник

Юрій ВІНОГРАДОВ

* Якщо визначені консультанти. Консультантом не може бути зазначено керівника дипломного проєкту.

АНОТАЦІЯ

У даній бакалаврській роботі були досліджені способи покращення взаємодії між робітниками та підрозділами підприємства за допомогою зворотного зв'язку. Були розглянуті засоби внутрішньої комунікації, що сприяють прозорому розвитку колективу, як команди. В результаті був створений програмний додаток на базі ОС Android для обміну зворотнім зв'язком та його обговоренням між працівниками компанії.

Створена програма дає можливість користувачам поділитися своєю позитивною або негативною думкою, зауваженнями та побажаннями стосовно будь-чого в компанії, починаючи з атмосфери в колективі та закінчуючи робочим процесом.

ABSTRACT

In this bachelor's thesis, ways to improve the interaction between employees and departments of the company through feedback were explored. The means of internal communication that promote the transparent development of the workers as a team were considered. As a result, an Android-based software application was created to share feedback and discuss it between company employees.

The created program allows users to share their positive or negative opinions, comments and wishes about anything in the company, from the atmosphere in the team and ending with the work process.

[illegible]

ТЕХНІЧНЕ ЗАВДАННЯ

**до дипломного проєкту
освітньо-кваліфікаційного рівня бакалавр**

на тему: “Система комунікацій з зворотнім зв’язком”

Київ – 2020 року

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ.....	2
2. ПІДСТАВИ ДЛЯ РОЗРОБКИ	2
3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ	2
4. ДЖЕРЕЛА РОЗРОБКИ	2
5. ТЕХНІЧНІ ВИМОГИ	2
5.1. Вимоги до програмного продукту, що розробляється.....	2
5.2. Вимоги до апаратного забезпечення мобільного пристрою	3
5.3. Вимоги до програмного забезпечення мобільного пристрою	3
6. ЕТАПИ РОЗРОБКИ.....	3

					ІАЛЦ.467100.002 ТЗ							
Зм.	Арк.	№ докум.	Підпис	Дата	Система комунікацій із зворотнім зв'язком Технічне завдання			Літ.	Аркуш	Аркушів		
Розробив	Куш Р.В.										1	3
Перевірив	Виноградов Ю.М.											
Реценз.												
Н. Контр.	Сімоненко В. П.											
Затвердив												
					НТУУ КПІ, ФІОТ, ІП-62							

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Найменування: «Система комунікацій з зворотнім зв'язком».

Область застосування: мобільний додаток може використовуватися в компаніях, які мають сформовану організаційну структуру та корпоративну адресу для авторизації співробітників в системі.

2. ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання бакалаврської дипломної роботи, затверджене кафедрою обчислювальної техніки Національного технічного університету України «Київський Політехнічний Інститут імені Ігоря Сікорського».

3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою розробки є створення програмного додатку для поширення зворотного зв'язку та його обговорення в межах компанії.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелами розробки є науково-технічна література, публікації в спеціалізованих виданнях та мережі Інтернет.

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до програмного продукту, що розробляється

Додаток, що розробляється, повинен:

- надавати можливість працівникам реєструватися та входити в систему за допомогою корпоративної пошти;

					ІАЛЦ.467100.001 ТЗ	Арк.
						2
Зм.	Арк.	№ докум.	Підпис	Дата		

- відправляти зворотній зв'язок з вибором потрібної групи одержувачів;
- отримувати та групувати зворотній зв'язок;
- надавати можливість коментувати зворотній зв'язок співробітників та підтримувати його лайком;
- надавати можливість відмічати емоціональний стан користувача та переглядати графіки його зміни.

5.2. Вимоги до апаратного забезпечення мобільного пристрою

- оперативна пам'ять об'ємом не менше 1 Гб;
- вбудована пам'ять об'ємом не менше 4 Гб;
- діагональ екрана не менше 4 дюймів.

5.3. Вимоги до програмного забезпечення мобільного пристрою

- версія ОС Android не менше 5.1;
- версія ОС IOS не менше 11.0.

6. ЕТАПИ РОЗРОБКИ

Назва етапів виконання	Термін виконання
Затвердження теми роботи	01.09.2019-01.01.2020
Вивчення та аналіз завдання	01.01.2020-28.02.2020
Аналіз існуючих передумов для розробки	28.02.2020-20.03.2020
Проектування архітектури системи	20.03.2020-25.04.2020
Розробка додатку	25.04.2020-22.05.2020
Оформлення матеріалів роботи	22.05.2020-26.05.2020
Передзахист	26.05.2020
Захист	

Пояснювальна записка

до дипломного проєкту

на тему: «Система комунікацій з зворотнім зв'язком»

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1. ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ.....	4
1.1 Комунікація в команді.....	4
1.2 Зворотній зв'язок як інструмент розвитку.....	6
1.3 Огляд існуючих систем.....	8
Висновки до розділу 1.....	15
РОЗДІЛ 2. АНАЛІЗ ФУНКЦІОНАЛУ СИСТЕМИ.....	16
2.1 Встановлення вимог та завдань.....	16
2.2 Опис функціоналу системи.....	17
2.3 Варіанти використання системи.....	20
2.4 Нефункціональні вимоги.....	35
Висновки до розділу 2.....	37
РОЗДІЛ 3. РЕАЛІЗАЦІЯ СИСТЕМИ.....	38
3.1 Вибір платформи для реалізації.....	38
3.1.1 Операційна система Android.....	39
3.1.2 Операційна система IOS.....	40
3.1.3 Мультиплатформна розробка.....	42
3.2 Компоненти серверної частини.....	43
3.2.1 Сервіс авторизації користувачів.....	44
3.2.2 База даних.....	45
3.2.3 Хмарні функції.....	48
3.2.4 Сервіс пуш-сповіщень.....	50

					ІАЛЦ.467100.003 ПЗ						
Зм.	Арк.	№ докум.	Підпис	Дата	Система комунікацій з зворотнім зв'язком Пояснювальна записка	Літ.			Аркуш	Аркушів	
Розробив	Куц Р.В.								1	72	
Перевірив	Виноградов Ю.М										
Реценз.											
Н. Контр.	Сімоненко В. П.										
Затвердив											
						НТУУ «КПІ ім. Ігоря Сікорського», ФІОТ, ІП-62					

3.3 Архітектура клієнтської частини.....	51
3.4 Компоненти клієнтської частини.....	54
Висновки до розділу 4.....	59
РОЗДІЛ 4. ІСТРУКЦІЯ КОРИСТУВАЧА.....	60
Висновки до розділу 4.....	69
ВИСНОВКИ.....	70
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	71

					ІАЛЦ.467100.003 ПЗ	Арк.
						2
Зм.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Позаду будь-якої компанії в світі стоїть команда, але не кожній організації вдається досягти максимального результату в спільній роботі над своїми проектами та стабільного розвитку колективу як єдиного механізму. Ні для кого не секрет, що не існує ідеальної групи людей між якими завжди присутнє повне взаєморозуміння та бачення кінцевого результату. Але даний факт є більш позитивним, якщо дивитися з боку прогресу. Постійна сутичка не схожих між собою думок і є тією рушійною силою, що направляє компанію до кращих результатів. Тому перед усім важливо налагодити максимально ефективний процес комунікації між командами та членами всередині кожної.

Організація росте та зміцнює свою позицію на ринку, якщо кожна її частина прагне постійного вдосконалення робочого процесу, взаємодії між колегами, умов в яких працюють люди, тощо. Найкращим способом вдосконалюватися є пошук своїх недоліків та робота над ними. Щоб зробити даний процес максимально швидким та ефективним, багато організацій дають можливість своїм працівникам ділитися зворотнім зв'язком. Але не у всіх даний процес є ефективним та автоматизованим.

Тому метою даної дипломної роботи є дослідження важливості зворотного зв'язку на покращення робочого процесу та створення програмного додатку, що допоможе організаціям вийти на новий рівень в розвитку внутрішньої взаємодії, автоматизувати збір особистої думки кожного співробітника та дати можливість висловити свою. Основна ідея – це створити процес комунікації навколо зворотного зв'язку, в якому може прийняти участь кожен, та дати працівникам можливість швидко та вільно поділитися особистою думкою стосовно всього, що відноситься до спільної роботи в компанії, без бюрократії та очікування регулярного збору зворотного зв'язку.

					ІАЛЦ.467100.003 ПЗ	Арк.
						3
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1

ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Комунікація в команді

Комунікація – це процес обміну інформацією, фактами, ідеями, пропозиціями, зауваженнями, поглядами, емоціями між двома або більше особами, за допомогою вербальних або невербальних засобів спілкування[1]. Як показано на рис. 1.1 комунікативний процес обов’язково вимагає участі не менше двох учасників: відправник(Sender) та отримувач(Receiver). Передача інформації являє собою кодування(Encoding) повідомлення(Message) із сторони відправника та розкодування(Decoding) з боку отримувача. В ході даного перетворення на передану інформацію впоиває шум(Noise). В ролі шуму може виступати все, що спотворює зміст повідомлення.

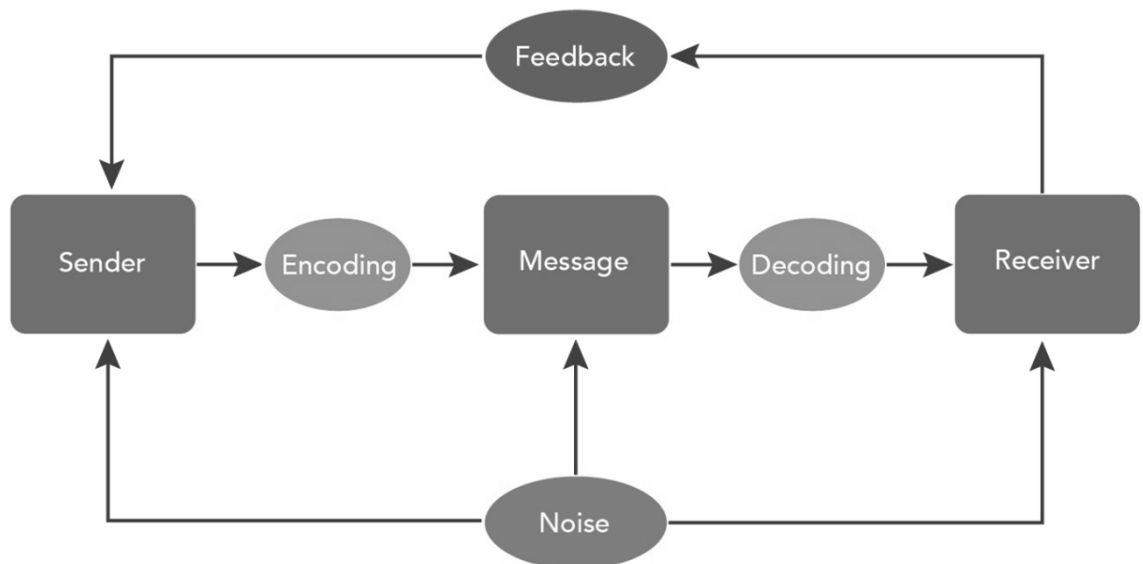


Рис. 1.1 Схема комунікативного процесу[2]

Обов’язковою складовою також є зворотній зв’язок(Feedback), який надає комунікації конструктивний характер. Комунікативний процес є невід’ємною частиною робочого циклу організації. Результат та успіх проекту напряду залежить від вміння людей працювати разом. Ця здатність будується за допомогою комунікативних навичок кожного робітника. Колегам слід створювати довірливі стосунки між собою с початку знайомста, тому що вони не з’являються самі по собі. Тож члени однієї команди повинні більш відкрито

обговорювати теми, які стосуються робочого процесу. В той же час всі обговорення повинні бути максимально чесними та відкритими з усіх сторін комунікативного процесу, інакше цілісність команди буде порушено. Товариші по команді завжди повинні ділитися інформацією та своїми почуттями друг з другом. Таким способом атмосфера довіри буде рости, а покращення робочого процесу не заставить себе довго чекати. Дослідження показують, що команди з 3 або більше членів є набагато ефективнішими ніж одна людина, яка тратить весь час на роботу. Але важливо відмітити, що за відсутності спілкування навіть команда найкращих професіоналів покаже гірший результат, а показники швидкості роботи значно зменшаться.

Ефективне спілкування допомагає чітко розподілити обов'язки та зрозуміти межі відповідальності всіх членів команди, що спонукає до зрозуміння того, що потрібно зробити. Команди без ефективного спілкування втрачають час, виконуючи роботу повторно або надаючи пріоритет не важливим завданням в конкретний момент часу. Також з'являється непорозуміння між співробітниками та хибна думка відносно інших членів команди, що зачасту є причиною виникнення конфлікту та поступовго зникнення довіри. В результаті кожен втрачає свою ефективність, адже з втратою зворотного зв'язку зникає і можливість рівнятися на інших. Кожен починає жити у своєму відокремленому від усіх світі та уявляти робочий процес по своєму. Тому в досягненні максимального успіху в роботі над проектом важливу роль відіграє правильно налаштована комунікація та готовність до неї кожного члена команди.

Важливо вміти правильно доносити свою думку до інших. Навики зрозуміло висловлювати свою думку незалежно від способу передачі інформації чи то усно, чи письмово, значно підвищують ефект спілкування. Адже ми повинні бути впевнені, що наші думки чітко та без змін дійшли до інших, інакше процес комунікації може стати деструктивним та принести більше проблем, ніж користі.

Так як інтернет спрощує та робить швидким процес обміну інформацією, більшість надають перевагу онлайн спілкуванню. Адже лише в

					ІАЛЦ.467100.003 ПЗ	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

онлайні є можливість бути весь час у курсі подій, якщо Ви маєте інтернет, який є достатньо доступний в наш час. Щоб отримати максимальний ефект від даного виду комунікації потрібно дотримуватися декількох правил[4]:

1. Не соромитися. Якщо Ви маєте питання, пропозицію, зацікавленість, то завжди пишіть про це;
2. Постійно беріть участь в обговореннях;
3. Використовуйте смайлики або гіфки, щоб чіткіше передати свої почуття та емоції;
4. Намагайтесь виражати думку коротко та зрозуміло;
5. Залишайте зворотний зв'язок;
6. Будьте толерантні стосовно думок інших;

1.2 Зворотний зв'язок як інструмент розвитку

Зворотний зв'язок, або як його ще прийнято називати Feedback – це один з найважливіших етапів у роботі з командою. Насамперед слід зазначити, що це комплекс вербальних та невербальних засобів спілкування між колегами з метою донесення якихось особистих думок, вражень або пропозицій. Саме тому, щоб робота була ефективною необхідно правильно давати фідбек своїм колегам. Схему зворотного зв'язку зображено на рис. 1.2.

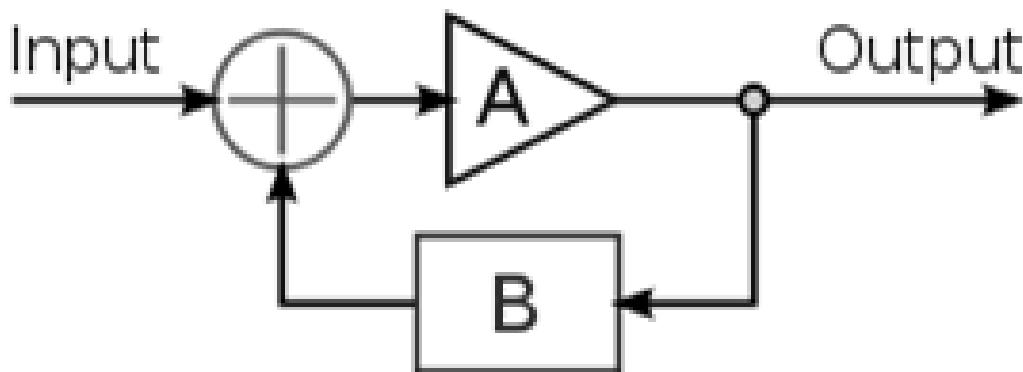


Рис. 1.2 Схема зворотного зв'язку[5]

До основних функцій фідбеку слід віднести те, що він: дає можливість відправнику повідомлення дізнатись, чи відбувся процес комунікації, значно інтерпретації повідомлень. Більшість психологів рекомендують навчитися слухати свого співрозмовника, це допоможе досягти максимального ефекту

від комунікативного процесу. Адже, чим менше ми будемо перебивати людину, роблячи їй зауваження, тим більш точно і правильно вона зможе донести свою думку.

Польський психолог Єжи Мелібруда зауважив, щоб досягти максимальної ефективності від зворотного зв'язку, його в першу чергу необхідно побудувати в формі описового ключа, а не в формі оцінок, тобто давати безоціночно і з дескриптивним характером. Це означає, що в більшій мірі необхідно говорити про свої спостереження, а не про висновки (завжди необхідно пам'ятати, що зроблені вами висновки можуть не відповідати дійсності). Можливо, що завдяки висловленим спостереженням партнер прийде до більш глибоких і правильних думок.

На відміну від деструктивного зворотного зв'язку, конструктивний є мотивацією для позитивної поведінки, яка підвищує продуктивність команди і при цьому зменшує негатив та напруження. Але зосереджуватися лише на позитиві та постійно вихвалити працівників з надією, що вони позбавляться своїх недоліків, – це невірний шлях, так само як і ніколи не визнавати досягнення своїх колег. Перші ніколи не зможуть знайти напрям для розвитку, а останні будуть тільки втрачати мотивацію. Тому конструктивний зворотний зв'язок – це перш за все золота середина між похвалою за досягнення та пропозиціями щодо подальшого вдосконалення. Щоб навчитися давати конструктивний зворотній зв'язок, потрібно завжди тримати в голові фактори, що роблять його таким, а саме:

1. Конкретність. Розмова повинна стосуватися конкретної теми;
2. Щирість. Потрібно чесно виражати свою думку та робити це без перебільшень;
3. Своєчасність. Подія повинна обговорюватися одразу після її виникнення;
4. Гарна підготовка. Кожна похвала або критика повинна бути чітко аргументована, тому варто підготуватися;

Згідно досліджень Гарвардської школи бізнесу[7], новачки більше прагнуть отримати позитивний зворотній зв'язок ніж експерти в тій чи іншій

					ІАЛЦ.467100.003 ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

сфері. Останні вважають, що негативний зворотній зв'язок – це гарний спосіб спрямувати свою діяльність на вдосконалення свої ефективності. На рис.1.3 зображено відсоткове співвідношення кількості опитуваних, які надають перевагу отриманню або надаванню позитивного та негативного зворотного зв'язку. Кількість учасників, що вважають кращим отримувати негативний зворотній зв'язок більша, ніж тих хто полюбляє більше похвалу. При цьому значний відсоток людей уникають вираження критики стосовно інших. Це говорить про те, що конструктивна критика краще впливає на подальший розвиток людини, ніж похвала. Але не всі люди готові вказувати іншим на їх недоліки.

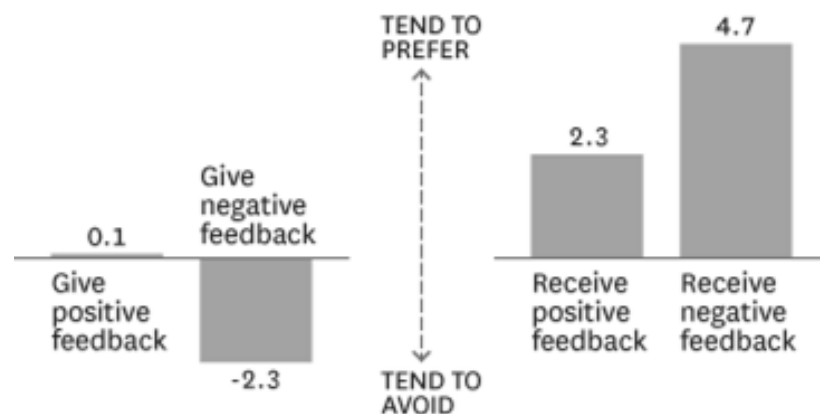


Рис. 2.3 Ставлення людей до зворотного зв'язку[7]

Підвищити кількість охочих ділитися негативним фідбеком можна за рахунок анонімності. Мало хто зважиться сказати в обличчя своєму начальнику все, що про нього думає. Тому дуже популярними стали анонімні форми фідбеку, за допомогою яких працівники чесно виражають свої думки без страху про можливі негативні наслідки.

1.3 Огляд існуючих систем

Slack – це месенджер, який призначений для корпоративного використання та з 2013 року об'єднує мільйони працівників по всьому світу. Вже давно дана система стала універсальним засобом комунікації для компаній. Основною метою для розробників даного сервісу було поєднання

багатьох інструментів, якими користуються організації в одну багатофункціональну систему. Таким чином Slack поєднав функції месенджера, файлообмінника та корпоративної пошти. З стрімким поширенням віддаленої від офісу роботи в систему було додано відео та аудіо зв'язок, щоб зробити співпрацю членів команди з різних частин світу більш зручною та ефективнішою. Slack почав своє існування, як інструмент для обміну повідомленнями в онлайн грі Glitch. Сьогодні в компанії працює понад 200 співробітників, які постійно додають новий функціонал, що полегшує та автоматизує робочий процес. Основною перевагою даного сервісу є можливість гнучкого поділу та групуванню штату компанії по каналах. Кожний відділ може сконцентруватися на власних завданнях. Slack також виділяє пам'ять своїх серверів для збереження всіх завантажених файлів та історії повідомлень. Тож кожен новий учасник команди може самостійно дізнатися про всі важливі пункти в роботі з історії повідомлень.

Переваги використання:

1. Можливість створення каналів за певною тематикою;
2. Зручна навігація між каналами;
3. Наявність версій під всі платформи: мобільний додаток для ОС Android та IOS, десктопний додаток та веб-додаток;
4. Підтримує аудіо та відео зв'язок;
5. Широкий спектр інтегрованих сторонніх систем;
6. Зберігає історію повідомлень;
7. Зберігає завантажені файли.

Недоліки використання:

1. Спілкування відбувається вільно без прив'язки до конкретної теми;
2. Обмежена безкоштовна версія;
3. Відсутність функціоналу для зворотного зв'язку.

					ІАЛЦ.467100.003 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

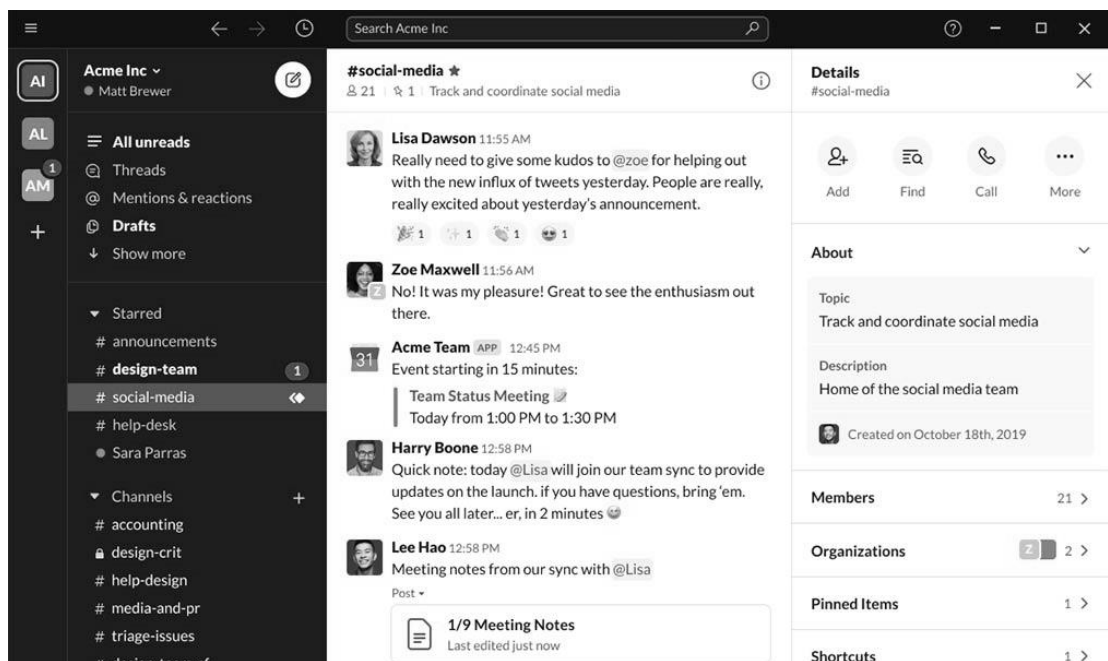


Рис. 1.4 Інтерфейс десктопної версії Slack[9]

Google Forms – це сервіс для проведення опитувань та збору інформації. Вперше даний інструмент був запущений компанією Google LLC в 2008 році. Основною перевагою Google Forms є простота у використанні та мінімальний інтерфейс для задуманого функціоналу. За допомогою даного сервісу можна швидко провести опитування серед працівників або клієнтів компанії, зібрати потрібну інформацію, зробити форму реєстрації на будь-яку подію та зібрати зворотний зв'язок стосовно конкретної теми. З новго функціоналу розробники додають нові способи та види опитувань. Інтуїтивно зрозумілий та легкий інтерфейс дозволяє швидко створити форму для збору потрібної інформації та розіслати її всім співробітникам.

Переваги використання:

1. Простий інтерфейс;
2. Синхронізація з сервісами Google;
3. Надає швидкий спосіб зібрати зворотний зв'язок;
4. Безкоштовний;
5. Наявність різних типів опитувань;
6. Гнучке налаштування форми.

Недоліки використання:

1. Потрібно самотійно відправляти запит співробітникам на заповнення форми;
2. Відсутня можливість обговорювати результати опитувань;
3. Результати опитувань приховані;
4. Працівники не можуть взяти на себе ініціативу дати фідбек, натомість повинні чекати на розсилку форми.

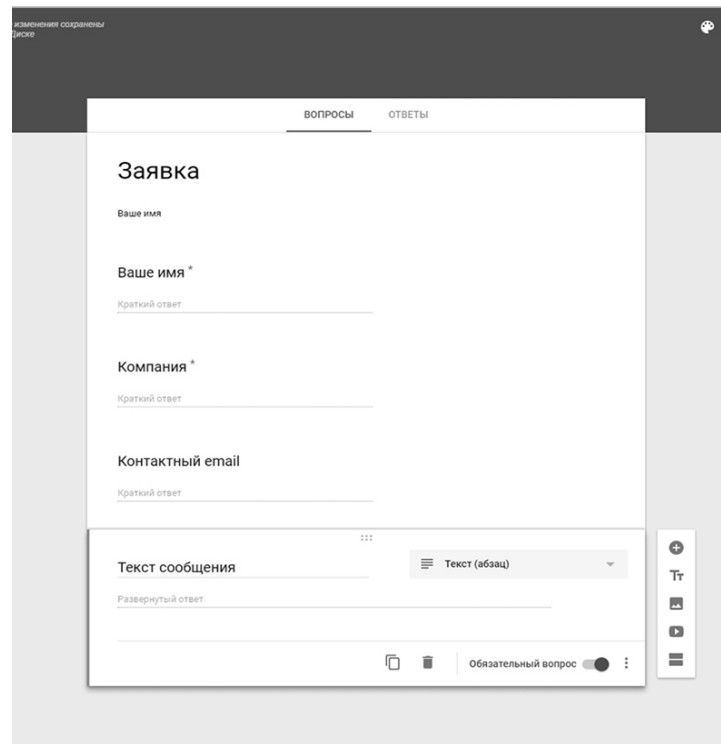


Рис. 1.5 Приклад Google форми[11]

Plai – онлайн сервіс для підвищення ефективності командної роботи та оцінки співробітників. Основними функціональними одиницями даного інструменту є планування цілей компанії та етапів розвитку на конкретні періоди часу, створення культури зворотнього зв'язку, відкритість перед усіма членами команди, система нагород за вдосконалення робочого процесу та досягнення цілей. Plai створений для розвитку таких цінностей компанії як прозорість, відповідальність, неперервне самовдосконалення. Основною ідеєю даного сервісу є впровадження системи OKR в процесі управління проектами. Даний підхід допомагає синхронізувати командні та індивідуальні цілі з впровадженням можливості контролювати процес досягнення

					ІАЛЦ.467100.003 ПЗ	Арк.
						11
Зм.	Арк.	№ докум.	Підпис	Дата		

поставлених задач всій команді.

Переваги використання:

1. Зручно та просто реалізована система OKR;
2. Можливість дати зворотний зв'язок стосовно співробітників;
3. Прозорість перед всіма членами команди;
4. Можливість інтегрувати сторонні сервіси;
5. Моніторинг продуктивності команди;
6. Наявність функціоналу для інших цілей;
7. Наявність мобільної версії.

Недоліки використання:

1. Відсутній вбудований функціонал для комунікації команди;
2. Зворотній зв'язок можна дати лише стосовно конкретного співробітника;
3. Обмежений функціонал в безкоштовній підписці;
4. Присутній визначений шаблон зворотного зв'язку;
5. Відсутність можливості коментувати зворотній зв'язок;
6. Низька популярність.

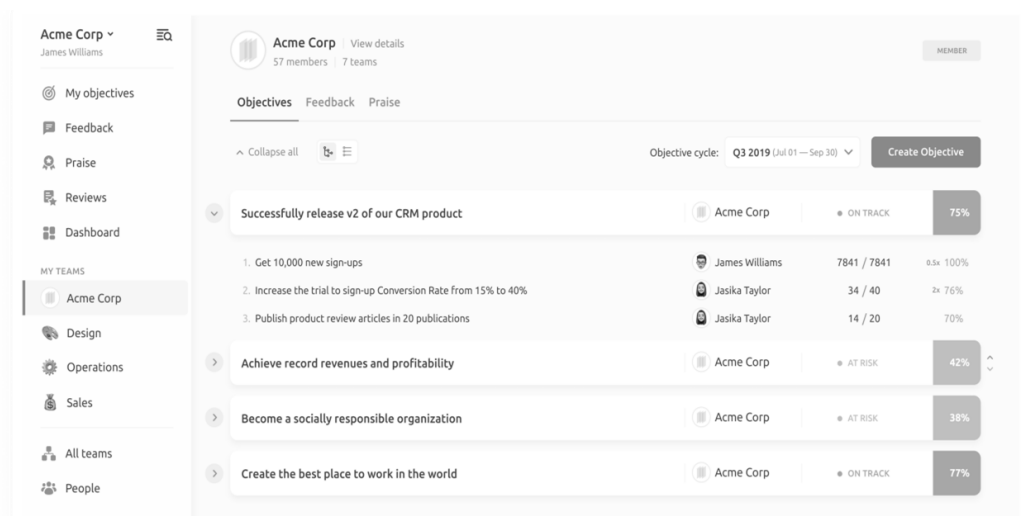


Рис. 1.6 Головна сторінка веб-додатку Plai[12]

Happy Job – це платформа для відстеження та покращення залученості персоналу в робочий процес. Платформа пропонує комплексне рішення для відстеження якості внутрішньої взаємодії персоналу. Даний сервіс дозволяє співробітникам компанії бути почутими. Збір зворотного зв'язку відбувається у вигляді регулярних коротких опитувань. Задля підвищення залученості до даного процесу працівників, питання представлені в легкій формі з великою варіацією типів. Даний підхід дає працівникам можливість відволіктися та трохи відпочити від інтенсивного робочого процесу з користю для компанії. Також даний продукт включає в себе постійний канал комунікації між кожним членом команди та управляючою ланкою. Адже в будь-який час кожен може поділитися своїми пропозиціями або критикою. Вся зібрана інформація аналізується та виставляється на загальне обговорення, таким чином кожен відчуває свою значущість, в результаті чого спільний рівень мотивації команди росте.

Переваги використання:

1. Прості та швидкі форми зворотнього зв'язку;
2. Співробітники мають можливість ділитися пропозиціями та критикою в будь-який момент;
3. Реалізовані версії для вебу та мобільних пристроїв;
4. Зворотний зв'язок має гнучкі налаштування отримувачів;
5. Наявність різноманітних форматів опитування.
6. Спрямований на покращення внутрішнього сервіса за допомогою залученості команди;

Недоліки використання:

1. Немає можливості відслідковувати пропозиції та критику співробітників в реальному часі;
2. Наявна лише платна версія з безкоштовним пробним періодом;
3. Відсутня можливість для обговорення командою зворотного зв'язку в режимі реального часу;

					ІАЛЦ.467100.003 ПЗ	Арк.
						13
Зм.	Арк.	№ докум.	Підпис	Дата		

ОБРАТНАЯ СВЯЗЬ

📅

📁

🖨

?

ПОДРАЗДЕЛЕНИЕ №1

СООБЩЕНИЯ 61

АРХИВ 14

ПЛАН РАБОТ 2

Вопросы: Есть ли у вас идеи, которыми хотите поделиться? ▼

Приоритет (Драйверы): Важные для критиков ▼

Теги / Метрики: ✖ Процессы, 📈 Изменения, Взаимодействие с коллегами ▼

Анонимно: Все ▼

Видит руководитель: Все ▼

Группа лояльности: Все ▼

СООБЩЕНИЕ	ТЕМЫ/ТЕГИ	УПР.
<div> <div></div> <div>Многие коллеги отдают планы в последний момент или позже условленного срока, что в свою очередь влечет за собой срывы по срокам предоставления IT-сервисов, а в нашем департаменте к этому привязаны размеры квартальных премий следующего ...</div> </div>	<div>✖ Процессы, 📈 Изменения</div>	<div>▼</div> <div>✕</div>
<div> <div></div> <div>Необходимо - разбивка задач по приоритетам, своевременное «планирование», дополнительные инструменты.</div> </div>	<div>Процессы компании</div>	<div>▲</div> <div>✕</div>
<div> <div>Вопрос: Есть ли у вас идеи, которыми хотите поделиться?</div> <div>Автор: Критик</div> <div>Создано: 27.09.2019</div> <div>Видит рук-ль: Да</div> </div>		
<div> <div></div> <div>Предлагаю в рамках ежегодной конференции провести футбольный турнир между регионами, организацию наш департамент готов взять на себя.</div> </div>	<div>👤 Коллеги, ❤ Баланс</div>	<div>▼</div> <div>✕</div>
<div> <div></div> <div>Добрый день. Нельзя ли как-то сократить ежедневные совещания? с 10 до 12 много звонков от клиентов, а мы вынуждены сидеть и слушать доклады других подразделений.</div> </div>	<div>👤 Руководитель</div>	<div>▼</div> <div>✕</div>
<div> <div></div> <div>Предлагаю централизованно провести обучение новых сотрудников работе с системой, тратим много времени на объяснения и помощь новичкам)</div> </div>	<div>Процессы компании</div>	<div>▼</div> <div>✕</div>

Рис. 1.7 Приклад зворотного зв'язку на платформі Harry Job[13]

Висновки до розділу 1

Комунікація та зворотній зв'язок є одними з найефективніших засобів підвищити мотивацію команди, покращити внутрішню атмосферу та робочий процес компанії. Злагоджена і націлена на результат команда швидше досягає успіху у своїх проектах. Тому пошук рішень для покращення комунікативних процесів з впровадженням зворотного зв'язку чи не одна з головних цілей багатьох компаній.

В першому розділі було розглянуто вплив комунікації та зворотного зв'язку на результати команди. Було досліджено ринок програмно забезпечення, що полегшує впровадження даних процесів в роботу компанії. Аналіз показав, що всі рішення мають свої недоліки, а єдиного програмного забезпечення, яке б допомагало побудувати ефективний процес комунікації навколо зворотнього зв'язку в режимі реального часу та дати користувачам можливість власноруч, в зручний для них час ділитися особистою думкою складно знайти у вигляді повноцінної системи.

					ІАЛЦ.467100.003 ПЗ	Арк.
						15
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2

АНАЛІЗ ФУНКЦІОНАЛУ СИСТЕМИ

Система призначена для корпоративного користування, тому основним завданням є створення зручного способу для комунікації, яка відбувається навколо зворотного зв'язку, між усіма працівниками компанії. Згідно з завданням ми поставили наступні цілі:

- поєднання функцій комунікації та зворотного зв'язку у зручній формі;
- створення ситеми із засобами відкритого поширення пропозицій та зауважень робітників в компанії;
- реалізація функціоналу для моніторингу емоційного стану співробітника.

Попередньо описані цілі дають можливість користувачу ділитися особистою думкою з усіма ланками компанії та відстежувати особистий емоційний стан для його аналізу. Зі сторони керівництва компанія отримує можливість відстежувати рівень задоволеності працівників, особисту думку кожного робітника стосовно покращення робочого процесу та реакцію колективу на пропозиції та зауваження.

2.1 Встановлення вимог та завдань

В системі можна виділити два актори: адміністратор та співробітник. В ролі адміністратора виступає компанія, а в ролі співробітника виступають всі користувачі, в тому числі і адміністратор. Актор користувач має спільний для адміністратора та співробітника функціонал.

Адміністратор має наступні можливості:

1. Реєстрація адміністратора;
2. Вхід в систему;
3. Створення простору компанії в системі;
4. Реєстрація співробітника в просторі компанії;
5. Перегляд всіх постів компанії;
6. Перегляд даних про настрій кожного користувача компанії.

					ІАЛЦ.467100.003 ПЗ	Арк.
						16
Зм.	Арк.	№ докум.	Підпис	Дата		

7. Видалення співробітника з системи.
8. Вихід із системи.

Співробітник має наступні можливості:

1. Вхід в систему;
2. Перегляд особистої інформації;
3. Налаштування звукових сповіщень;
4. Створення посту;
5. Перегляд існуючих постів;
6. Стверення коментарю до посту;
7. Оцінка посту та коментарю;
8. Оцінка теперешнього настрою;
9. Відображення динаміки настрою;
10. Вихід із системи.

Основні вимоги до системи:

1. Простий та зручний дизайн;
2. Можливість використовувати в будь-якому місці;
3. Реєстрація та авторизація відбувається за допомогою аккаунту Google;
4. Оновлення та отримання постів та коментарів в реальному часі;
5. Сповіщення про отримання нових постів та коментарів;
6. Можливість відправляти зворотній зв'язок окремим категоріям співробітників;
7. Можливість створення позитивного і негативного зворотного зв'язку;
8. Можливість перегляду постів по категоріям;
9. Можливість анонімно відправляти зворотній зв'язок;
10. Можливість відстежувати зміну загального настрою всіх користувачів;
11. Можливість відстежувати зміну настрою користувача.

2.2 Опис функціоналу системи

1. Функціонал системи розподілений на дві частини: додаток для співробітників та адміністративна панель для керування системою зі сторони компанії. Для належного функціонування, система повинна

					ІАЛЦ.467100.003 ПЗ	Арк.
						17
Зм.	Арк.	№ докум.	Підпис	Дата		

надавати співробітникам та адміністратору наступні можливості використання: Реєстрація адміністратора. Адміністратор може зареєструватися в системі за допомогою Google акаунту. В результаті адміністратор отримає доступ до адміністративної панелі. Створення простору компанії в системі. Адміністратор має можливість створити в системі простір для компанії. Для цього потрібно ввести всю необхідну інформацію про компанію та налаштувати організаційну структуру компанії;

2. Реєстрація співробітника в просторі компанії. Реєстрація співробітника повинна відбуватися за допомогою Google аккаунту. Адміністратор повинен створити опублікований запис співробітника в просторі компанії, створеного в системі, та прив'язати пошту Google аккаунту до нього.
3. Вхід у систему. Співробітник або адміністратор може увійти в систему за допомогою аккаунту Google. В результаті система відкриє екран, відповідно до прав доступу користувача.
4. Перегляд особистої інформації співробітника. Співробітник на екрані профілю має можливість переглядати наступні дані свого опублікового запису в системі:
 - Фото опублікового запису;
 - ФІО;
 - Корпоративну пошту;
 - Назву відділу, в якому працює;
 - ФІО свого керівника;
5. Налаштування звукових сповіщень. Користувач на екрані профілю може задавати наступні налаштування для сповіщень системи:
 - а. Час сповіщення, яке нагадує відмітити свій настрій;
 - б. Включити або виключити сповіщення про нові пости;
 - с. Включити або виключити сповіщення про нові коментарі до свого посту.
6. Створення нового посту. Співробітник може створити новий пост з

					ІАЛЦ.467100.003 ПЗ	Арк.
						18
Зм.	Арк.	№ докум.	Підпис	Дата		

7. вибором наступних критеріїв: позитивний або негативний зворотній зв'язок
8. Категорія співробітників, які зможуть переглядати пост. Існує 4 категорії одержувачів посту:
- a. Всі співробітники компанії;
 - b. Всі співробітники відділу, в якому працює користувач;
 - c. Всі співробітники будь-якого іншого відділу компанії;
 - d. Керівник відділу, в якому працює користувач, та голова правління;
9. Показувати іншим співробітникам автора поста чи ні;
10. Перегляд існуючих постів. Користувач може переглянути пости, якщо він входить в категорію одержувачів цього посту. Існує 5 категорій перегляду посту:
- a. «Вся компанія» – пости, які були відправлені всім співробітникам компанії;
 - b. «Мій відділ» – пости, які були відправлені співробітниками з відділу, в якому працює користувач, в адресу цього ж відділу;
 - c. «Інший відділ» – пости, які були відправлені в адресу нашого відділу від співробітників з інших відділів, та пости, які були відправлені в адресу будь-якого іншого відділу від співробітника з відділу, в якому працює користувач;
 - d. «Мені» – пости, які були відправлені в адресу керівника відділу від співробітників цього ж відділу, якщо користувач є керівником відділу, та пости, які були відправлені в адресу керівника підлеглого користувачу відділу, якщо користувач є головою правління;
 - e. «Мої» – всі пости, які відправив користувач.
11. Створення коментарю до посту. Співробітник може коментувати видимі йому пости на екрані деталей посту.
12. Оцінка посту та коментарю. Співробітник може оцінити пост або коментар, натиснувши на кнопку «Подобається». В результаті

					ІАЛЦ.467100.003 ПЗ	Арк.
						19
Зм.	Арк.	№ докум.	Підпис	Дата		

кількість співробітників, яким сподобався пост або коментар збільшиться на один. У разі повторного кліку на кнопку «Подобається» кількість співробітників, яким подобається пост або коментар, зменшиться на один;

13. Оцінка теперішнього настрою співробітника. На екрані «Настрій» співробітник може обрати та зберегти свій теперішній настрій з п'яти запропонованих варіантів за шкалою від 1 до 5;

14. Відображення динаміки настрою співробітника. На екрані «Графік» співробітник може переглядати графік зміни середнього значення настроїв кожного співробітника компанії за останній тиждень та діаграму відношень різних значень свого настрою за останній місяць;

15. Вихід із системи. На екрані «Профіль» співробітник або адміністратор може вийти з системи. В результаті він буде перенаправлений на екран входу.

2.3 Варіанти використання системи

Нижче наведені можливі варанти використання системи з боку адміністратора та користувача. Для адміністратора можливі наступні сценарії використання системи.

Сценарій реєстрації адміністратора в системі(Рис.2.1)

Передумови:

- Користувач має Google акаунт.

Успішний сценарій:

1. Користувач нажимає на кнопку зареєструватися;
2. Система відображає список авторизованих в телефоні Google акаунтів;
3. Користувач обирає акаунт для реєстрації;
4. Система реєструє адміністратора на основі обраного Google акаунту;
5. Система відкриває панель адміністратора.

Результат: адміністратор зареєстрований в системі та знаходиться на головному екрані адміністративної панелі.

					ІАЛЦ.467100.003 ПЗ	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

Виключні ситуації: адміністратор з обраним Google акантом вже існує в системі, жодного Google акаунту не авторизовано в телефоні.

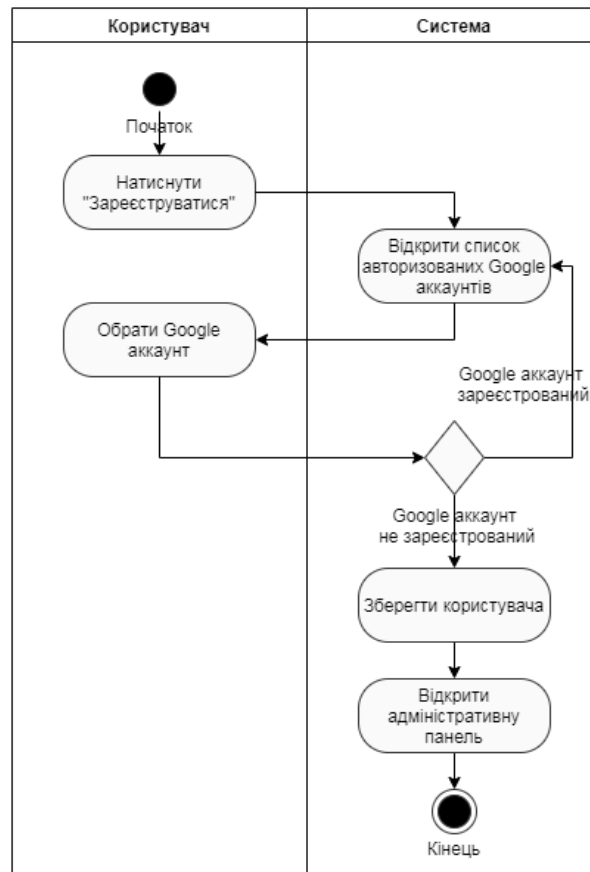


Рис. 2.1 Реєстрація адміністратора

Сценарій створення простору компанії в системі(Рис. 2.2)

Передумови:

- адміністратор зареєстрований в системі.

Успішний сценарій:

1. Адміністратор натискає на кнопку «Створити новий простір для компанії»;
2. Система відкриває форму створення новго простору для компанії;
3. Адміністратор заповнює форму та натискає «Створити»;
4. Система відкриває форму налаштування організаційної структури;
5. Адміністратор налаштовує організаційну структуру компанії та натискає кнопку «Зберегти»;
6. Система створює та відкриває простір компанії.

Результат: в системі створено простір для нової компанії з усією інформацією про компанію, введеною адміністратором.

Виключні ситуації: адміністратор не заповнив усі необхідні поля форми створення новго простору для компанії, створений простір уже існує в системі.



Рис. 2.2 Створення простору компанії

Сценарій реєстрації співробітника в системі(Рис. 2.3)

Успішний сценарій:

1. Адміністратор створює Google аккаунт користувачу;
2. Адміністратор обирає в системі пункт «Додати нового співробітника»;
3. Система відкриває форму заповнення інформації про співробітника;
4. Адміністратор заповнює форму;
5. Адміністратор натискає кнопку «Зберегти»;
6. Система зберігає нового користувача;
7. Система виводить повідомлення про успішне додавання нового користувача.

Результат: користувач зареєстрований в системі та може авторизуватися в ній.

Виключні ситуації: користувач з таким Google акаунтом вже існує, введені невірні дані, не заповнені всі необхідні поля форми реєстрації.



Рис. 2.3 Реєстрація співробітника

Сценарій авторизації користувача в системі(Рис.2.4)

Передумови:

- Користувач має Google акаунт з корпоративною поштою;
- Google акаунт користувача зареєстрований в системі.

Успішний сценарій:

1. Користувач натискає на кнопку «Увійти»;
2. Система відображає список авторизованих в телефоні Google акаунтів;
3. Користувач обирає Google акаунт для авторизації;
4. Система авторизує користувача;
5. Система відкриває головний екран додатку, відповідно до прав доступу користувача;

Результат: користувач увійшов в систему.

Виключні ситуації: Google акаунт користувача не зареєстрований в системі.

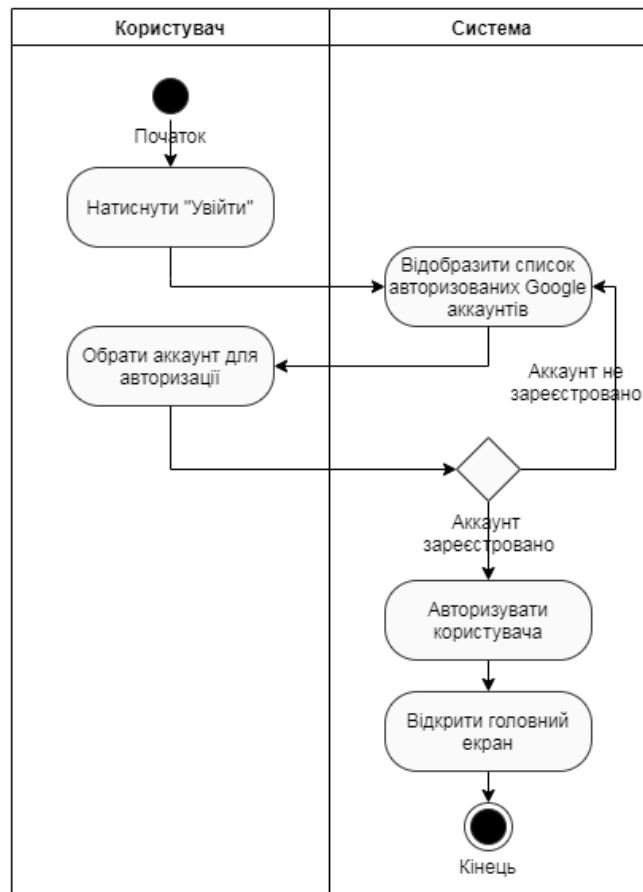


Рис. 2.4 Авторизація користувача

Сценарій перегляду всіх постів системи(Рис. 2.5)

Передумови:

- Адміністратор авторизований в системі.

Успішний сценарій:

1. Адміністратор вибирає простір компанії зі списку;
2. Система відкриває екран управління обраною компанією;
3. Адміністратор натискає на кнопку «Переглянути пости»;
4. Система відкриває екран з налаштуваннями фільтрів відображення постів;
5. Адміністратор налаштовує фільтри пошуку постів;
6. Система відображає список відфільтрованих постів.

Результат: система відображає адміністратору список постів із заданими фільтрами.

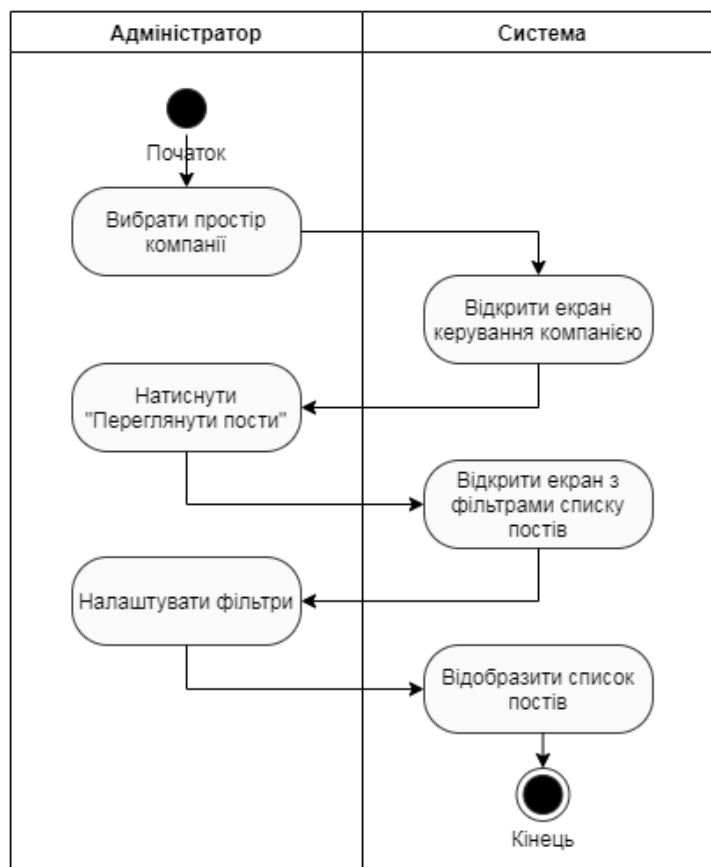


Рис. 2.5 Перегляд всіх постів в компанії

Сценарій перегляду динаміки настрою співробітників компанії(Рис. 2.6)

Передумови:

- Адміністратор авторизований в системі.

Успішний сценарій:

1. Адміністратор обирає простір компанії із списку;
2. Система відкриває екран управління обраною компанією;
3. Адміністратор натискає на кнопку «Переглянути динаміку настрою»;
4. Система відкриває екран з налаштуваннями фільтрів відображення графіків настрою;
5. Адміністратор налаштовує фільтри відображення значень настроїв;
6. Система відображає відфільтровані графіки настрою;

Результат: система відображає графіки настрою залежно від обраних фільтрів.

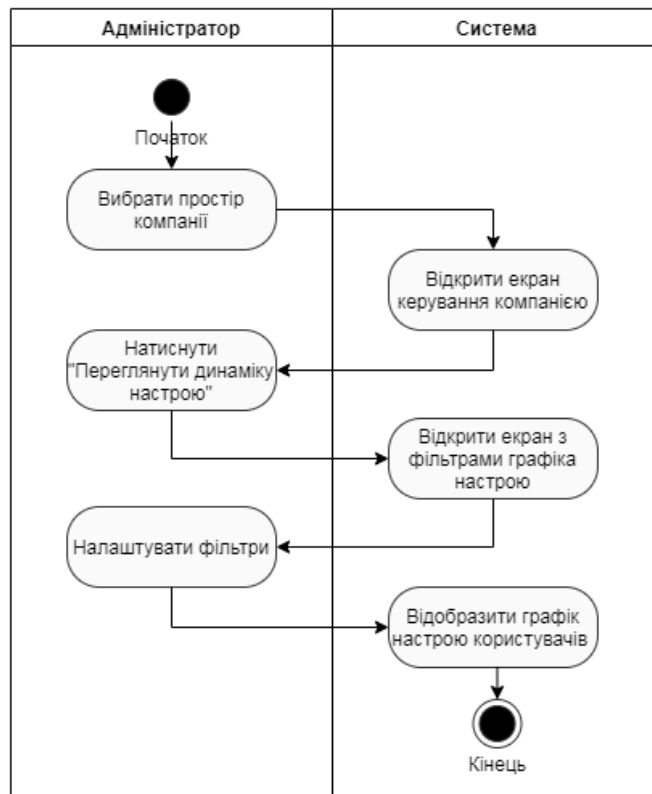


Рис. 2.6 Перегляд динаміки настрою компанії

Сценарій видалення співробітника із системи(Рис. 2.7)

Передумови:

- Адміністратор авторизований в системі;
- До простору компанії в системі був доданий мінімум один співробітник.

Успішний сценарій:

1. Адміністратор обирає простір компанії із списку;
2. Система відкриває екран управління обраною компанією;
3. Адміністратор натискає на кнопку «Співробітники»;
4. Система відкриває екран зі списком співробітників компанії;
5. Адміністратор обирає співробітника;
6. Система відкриває екран з інформацією про співробітника;
7. Адміністратор натискає кнопку «Видалити співробітника»;
8. Система видалляє всі дані співробітника;

Результат: система видалляє опублікований запис співробітника та блокує можливість входу для співробітника.

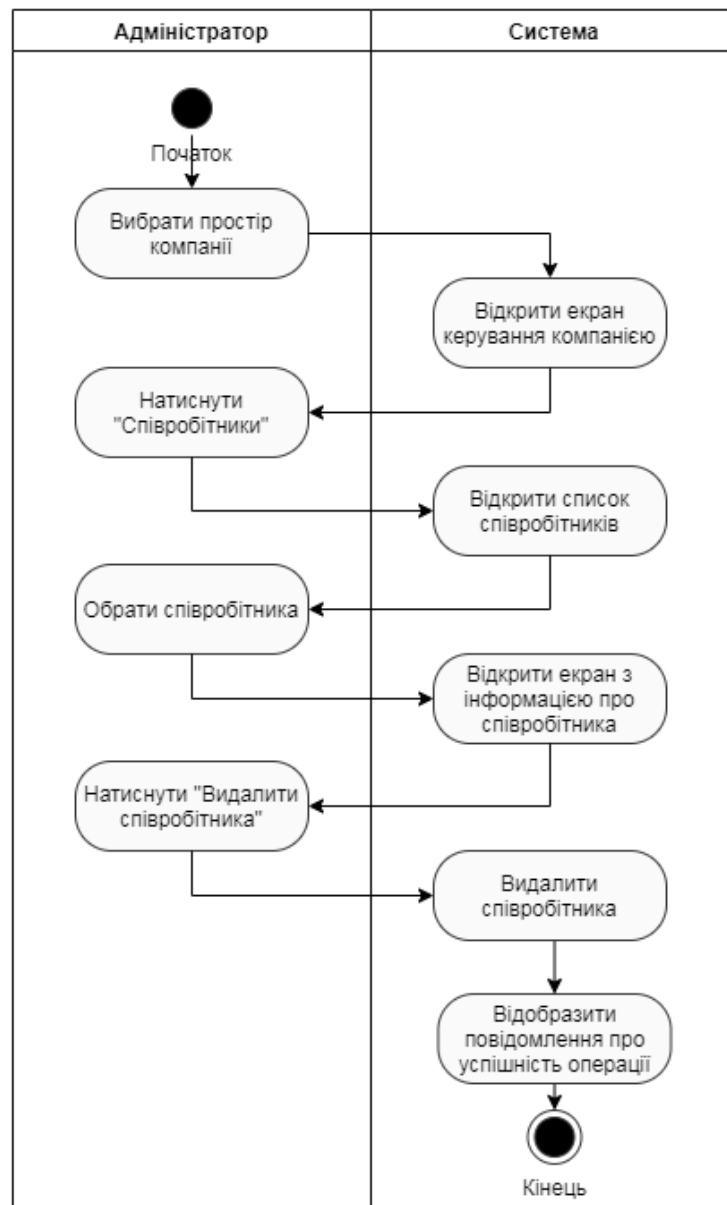


Рис. 2.7 Видалення співробітника

Сценарій перегляду особистої інформації(Рис. 2.8)

Передумови:

- Співробітник авторизований в системі.

Успішний сценарій:

1. Співробітник натискає на кнопку «Профіль»;
2. Система відкриває екран профілю;
3. Система відображає особисті дані користувача;

Результат: система відкриває екран профілю, на якому присутні особисті дані співробітника.

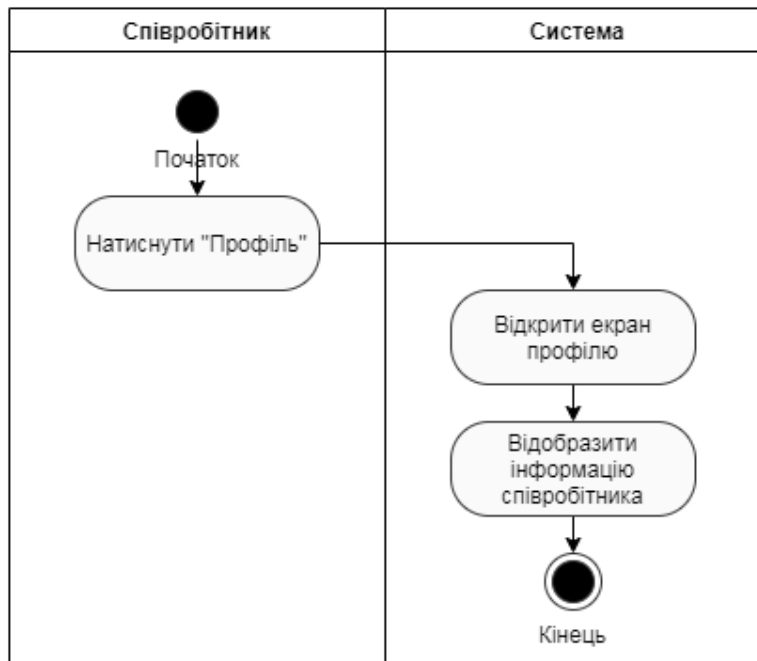


Рис. 2.8 Перегляд особистої інформації

Сценарій налаштування звукових сповіщень(Рис. 2.9)

Передумови:

- Співробітник авторизований в системі.

Успішний сценарій:

1. Співробітник натискає на кнопку «Профіль»;
2. Система відкриває екран профілю;
3. Співробітник вмикає або вимикає сповіщення про нові пости;
4. Система зберігає зміни;
5. Співробітник смикає або вимикає сповіщення про нові коментарі до його постів;
6. Система зберігає зміни;
7. Співробітник вводить час сповіщення про нагадування відмітити настрій;
8. Система зберігає зміни;

Результат: співробітник налаштовує роботу сповіщень, система миттєво зберігає зміни та змінює налаштування сповіщень для конкретного співробітника.

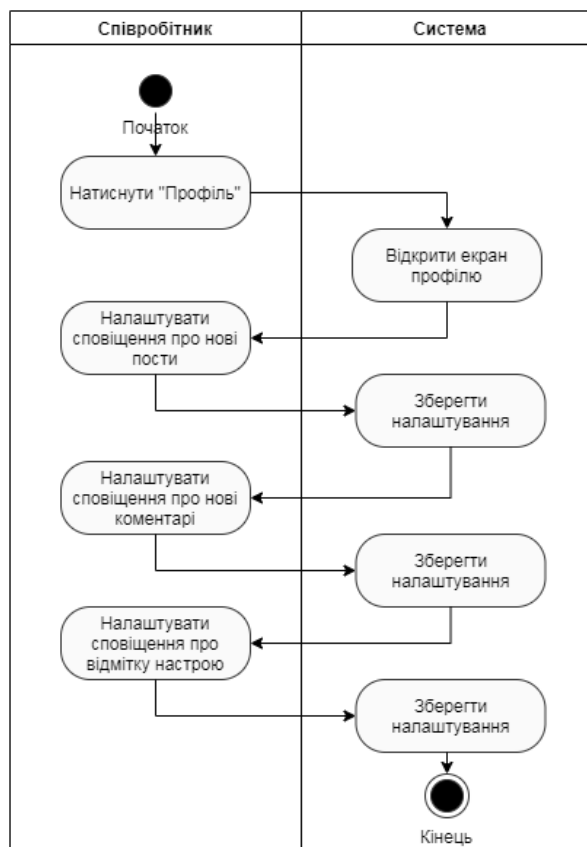


Рис. 2.9 Налаштування сповіщень

Сценарій створення посту(Рис. 2.10)

Передумови:

- Співробітник авторизований в системі.

Успішний сценарій:

1. Співробітник натискає на вкладку «Бомба»;
2. Система відкриває екран з вибором типу зворотного зв'язку;
3. Співробітник натискає на тип зворотного зв'язку, який хоче відправити;
4. Система відкриває екран з формою зворотного зв'язку;
5. Співробітник заповнює форму зворотного зв'язку;
6. Співробітник натискає кнопку «Відправити»;
7. Система зберігає пост;
8. Система надсилає сповіщення одержувачам.

Результат: пост збережений в системі, всім одержувачам було надіслано сповіщення про його створення.

Виключні ситуації: співробітник не ввів текст зворотного зв'язку.

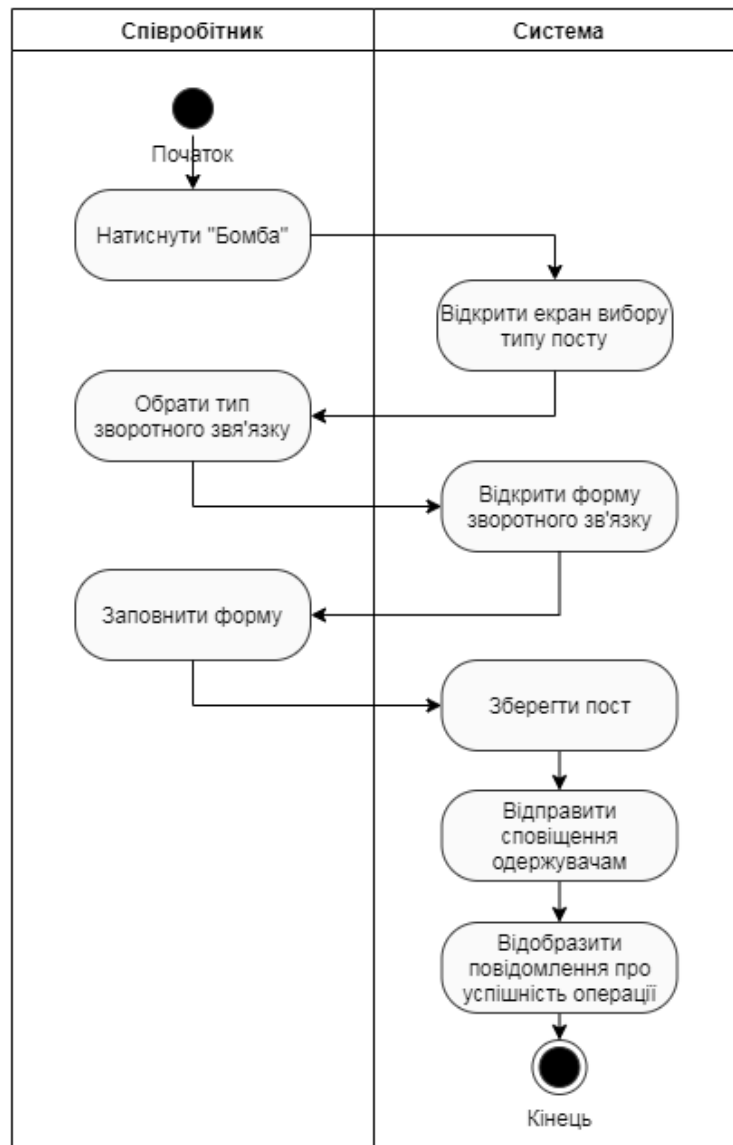


Рис. 2.10 Створення посту

Сценарій перегляду постів(Рис. 2.11)

Передумови:

- Співробітник авторизований в системі.

Успішний сценарій:

1. Співробітник натискає на вкладку «Обговорення»;
2. Система відкриває екран з списком постів, адресованих всій компанії;
3. Співробітник обирає вкладку зі списком постів певної категорії;
4. Система змінює відображений список на той, що обрав співробітник.;
5. Співробітник натискає на пост зі списку;
6. Система відкриває екран з деталями посту та коментарями до нього.

Результат: відкрито екран деталей посту з списком коментарів до нього.

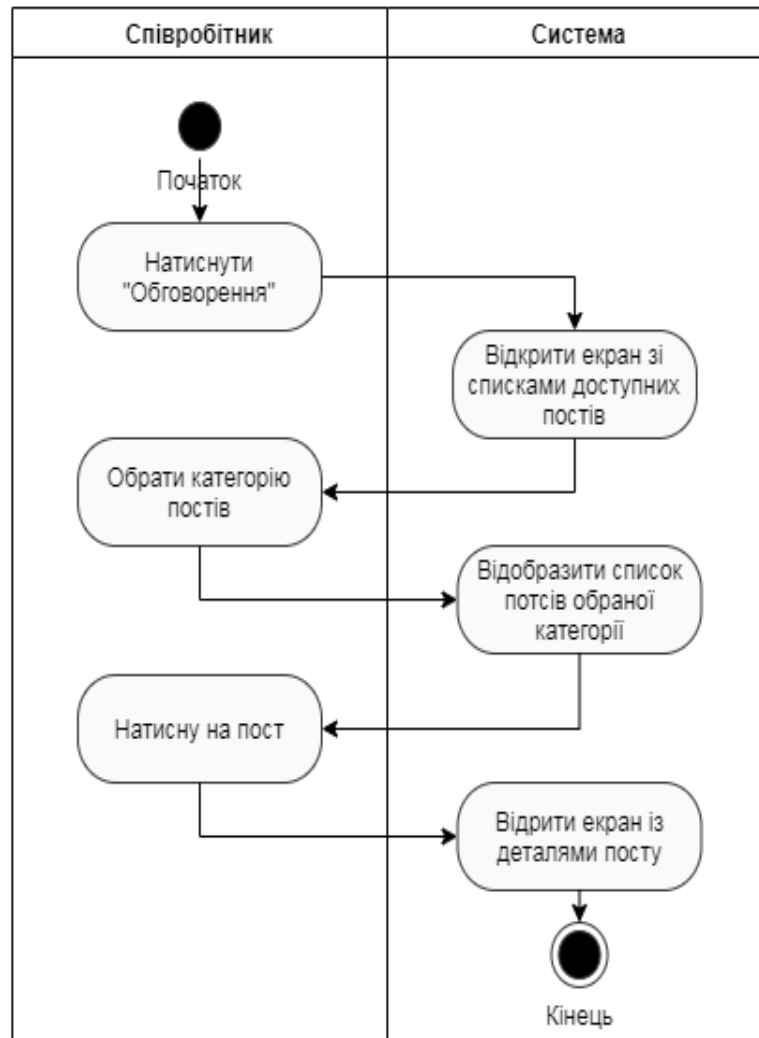


Рис. 2.1 Перегляд посту

Сценарій оцінювання теперішнього настрою(Рис. 2.12)

Передумови:

- Співробітник авторизований в системі.

Успішний сценарій:

1. Співробітник натискає на вкладку «Настрій»;
2. Система відкриває екран з вибором настрою;
3. Співробітник обирає свій настрій;
4. Співробітник закриває екран вибору настрою;
5. Система зберігає останній обраний настрій.

Результат: настрій збережено в системі, система перераховує середнє значення настрою всієї компанії.

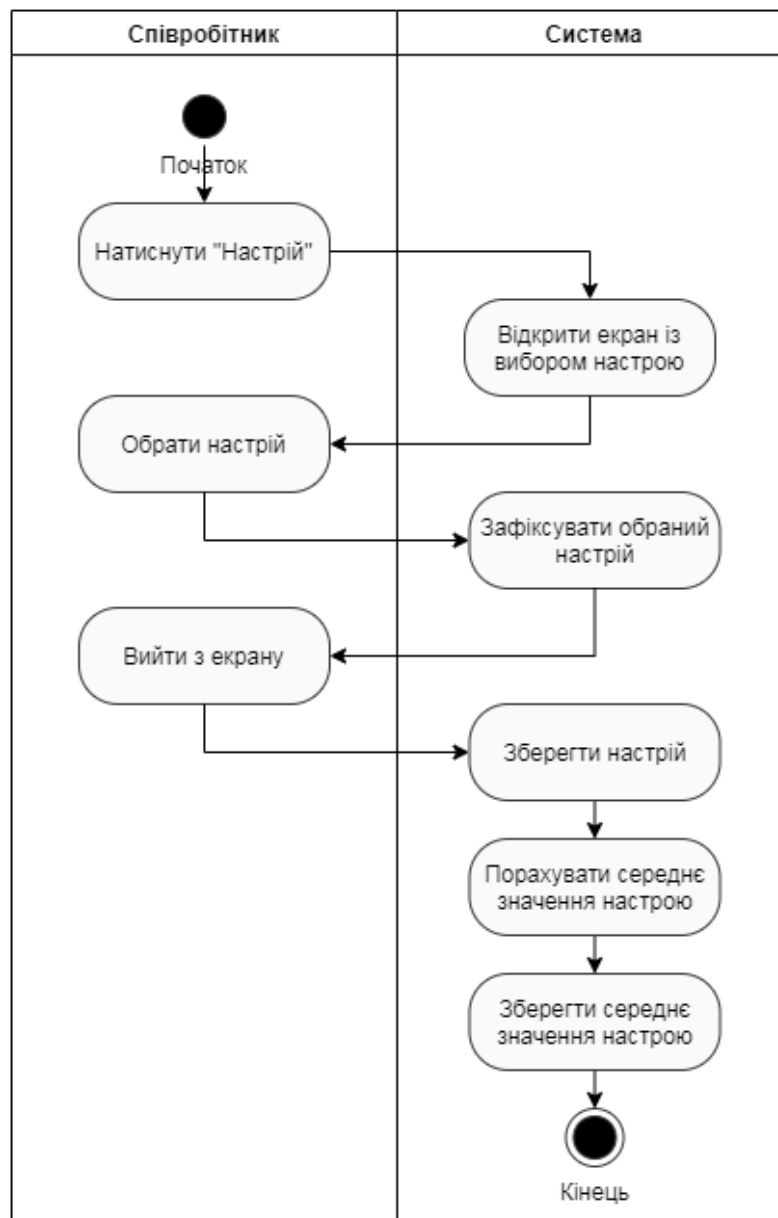


Рис. 2.12 Оцінювання теперішнього настрою

Сценарій перегляду динаміки настрою співробітника(Рис. 2.13)

Передумови:

- Співробітник авторизований в системі.

Успішний сценарій:

1. Співробітник натискає на вкладку «Графік»;
2. Система відкриває екран з графіками настрою.

Результат: система відображає графік середнього настрою в компанії та діаграму відсоткового відношення значень настрою співробітника за останній місяць.

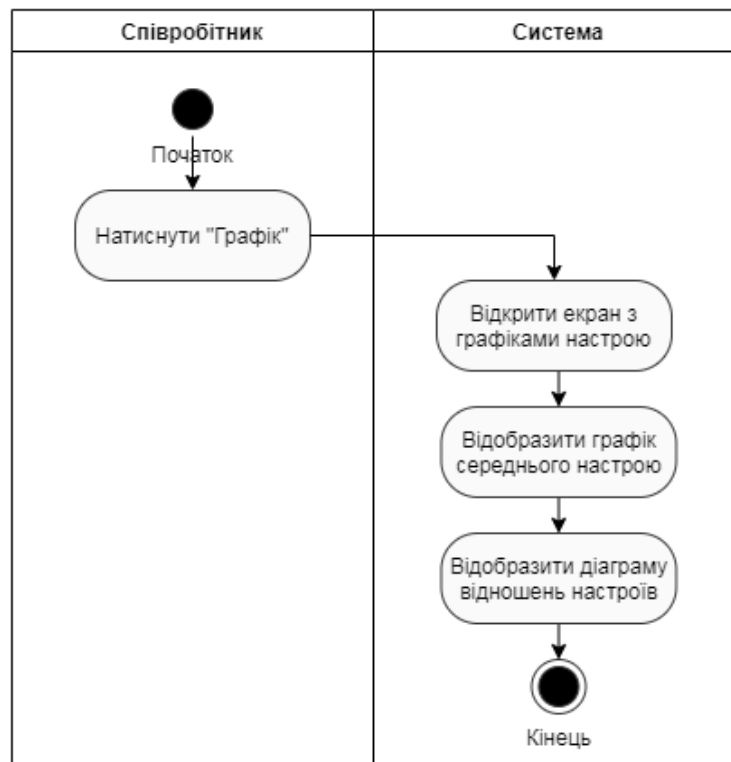


Рис. 2.13 Перегляд динаміки настрою

Сценарій створення коментарю до посту(Рис. 2.14)

Передумови:

- Співробітник авторизований в системі.

Успішний сценарій:

1. Співробітник натискає на вкладку «Обговорення».
2. Система відкриває екран зі списком постів.
3. Співробітник натискає на пост зі списку.
4. Система відкриває екран з деталями посту та коментарями до нього.
5. Співробітник вводить текст коментарю.
6. Співробітник натискає кнопку «Відправити».
7. Система зберігає коментар.
8. Система надсилає сповіщення автору посту.

Результат: коментар до посту збережено в системі, власник посту отримав сповіщення про новий коментар.

Виключні ситуації: співробітник натиснув кнопку «Відправити» з відсутнім текстом коментарю.



Рис. 2.14 Створення коментарю до посту

Сценарій виходу із системи(Рис. 2.15)

Передумови:

- Співробітник авторизований в системі.

Успішний сценарій:

1. Співробітник натискає на кнопку «Профіль».
2. Система відкриває екран профілю.
3. Співробітник натискає на кнопку «Вийти».
4. Система перенаправляє співробітника на екран авторизації.

Результат: співробітник вийшов із системи та був пере направлений на екран авторизації.

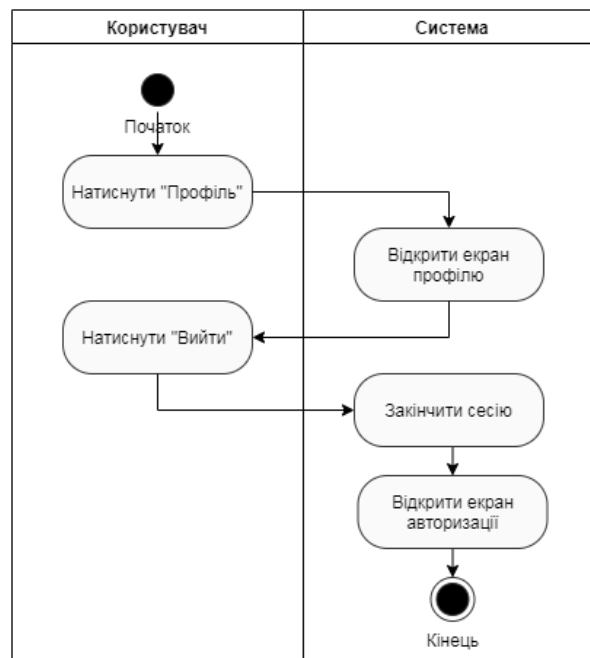


Рис. 2.15 Вихід із системи

2.2 Нефункціональні вимоги

Будь-яка система повинна працювати якісно та без критичних помилок. Тому поряд з функціональними вимогами стоять нефункціональні. Нефункціональні вимоги визначають якість роботи системи. До них відносяться наступні критерії: надійність, швидкість, безпека, адаптивність, масштабованість, гнучкість. Кожна функціональна вимога включає в себе увесь ряд нефункціональних. Адже від того наскільки система буде швидко реагувати, як вона буде працювати у різних станах платформи, на якій реалізована, наскільки користувачам буде зручно нею користуватися, залежить існування самої системи.

В сучасному світі конкуренція набирає стрімких обертів, тому для бізнесу дуже важливо швидко адаптуватися до змін на ринку. Постійний процес вдосконалення сервісів став обов'язковою частиною розробки кожного програмного забезпечення. Саме правильне визначення нефункціональних вимог з врахуванням можливих змін у майбутньому допомагає досягти стабільного розвитку системи.

Нефункціональні вимоги завжди впливають на реалізацію кожного модуля або компонента системи. Тому, щоб обрати оптимальні між часом, ресурсами та якістю рішення для реалізації системи комунікацій із зворотнім,

ми встановили наступні нефункціональні вимоги:

1. Система повинна давати доступ до всієї інформації адміністратору;
2. Система повинна бути доступна для швидкого використання в будь-якому місці;
3. Система повинна працювати без з'єднання з інтернетом;
4. Система повинна працювати цілодобово;
5. Система повинна виконувати операції не довше 5 сек;
6. Система повинна обробляти усі виключні ситуації;
7. Система повинна бути готовою до швидкої інтеграції з сторонніми сервісами;

					ІАЛЦ.467100.003 ПЗ	Арк.
						36
Зм.	Арк.	№ докум.	Підпис	Дата		

Висновки до розділу 2

У даному розділі було виділено основні вимоги до реалізації системи комунікацій із зворотнім зв'язком та завдання, які стоять перед нами, для досягнення максимальної ефективності у використанні нею. Були сформовані функціональні та нефункціональні вимоги до системи. На основі цих вимог було описано основні сценарії взаємодії користувача з системою. Щоб досягти максимальної ефективності та адаптивності роботи системи, було виділено два типи користувачів та описано їх основні способи та цілі вискоритскання системи. Було поставлене основне завдання: створити систему, яка б була легкою у використанні, працювала у режимі реального часу та надавала користувачам можливість ділитися своїми зауваженнями, пропозиціями та зворотнім зв'язком з можливістю його обговорення в подальшому .

					ІАЛЦ.467100.003 ПЗ	Арк.
						37
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3

РЕАЛІЗАЦІЯ СИСТЕМИ

3.1 Вибір платформи для реалізації

В ролі платформи для реалізації було обрано мобільний пристрій. Відповідно до функціональних та нефункціональних вимог, мобільний додаток надає можливість використовувати систему в зручний та потрібний для цього час, отримувати дані та сповіщення в режимі реального часу та має можливість для роботи в фоновому режимі. Тож можливості, які надають мобільні операційні системи надають функціонал та програмні можливості для реалізації системи комунікацій із зворотнім зв'язком. При цьому концепція мобільних додатків дозволяє придержуватися усім вимогам, які стоять перед системою, для її ефективної та коректної роботи.

Найбільш поширеними операційними системами, на яких працюють мобільні пристрої, є IOS та Android. Вони займають майже увесь ринок мобільних телефонів в 2020 році. Навіть при умові, що ОС IOS займає значно меншу частку ринку, велика кількість людей використовує її в ролі основної. Тому щоб покрити простір усіх потенційних користувачів система повинна бути реалізована на обох платформах.

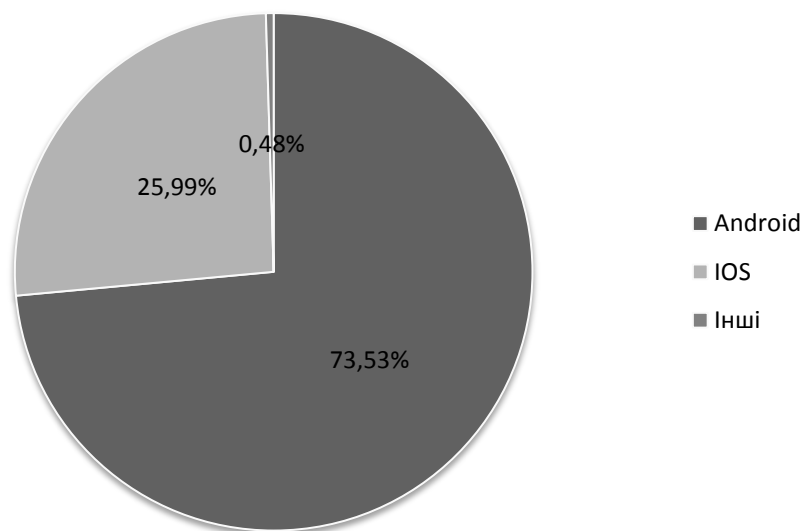


Рис. 3.1 - Частка використання мобільних операційних систем[14]

3.1.1 Операційна система Android

Ідея створити операційну систему Android з'явилася в компанії Android Inc. В 2005 році компанія Google купила Android Inc. 23 вересня 2008 року компанія випустила першу версію мобільного телефону Android. Так як дана система є відкритою, то 84 компанії об'єдналися в альянс для створення мобільних телефонів на базі операційної системи Android. Через те, що розробкою смартфонів займається велика кількість компаній по всьому світу, мобільні телефони з Android мають різноманітну лінійку вибору та є найпопулярнішими поміж інших мобільних операційних систем[15].

Операційна система Android працює на базі ядра Linux. Всі додатки працюють за допомогою JVM. Google створила власну версію JVM під назвою DALVIK, яка виконує усі додатки у вигляді байт-коду. Для цього був створений формат установочних пакетів .apk, для запуску програм на операційній системі Android. Починаючи з 2008 року Google випустила 10 версій операційної системи.

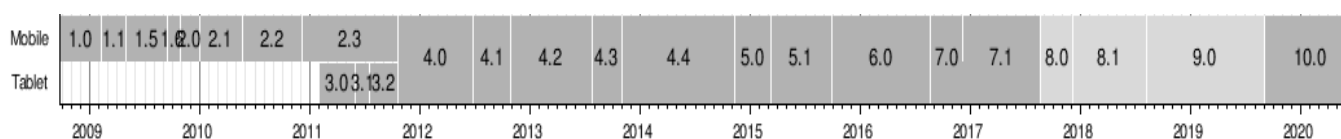


Рис. 3.2 – Версії операційної системи Android[16]

Основною мовою для створення додатків для ОС Android є Java. У 2009 році Google опублікувала Android Native Development Kit - пакет інструментаріїв і бібліотек, для запуску програмного коду на мові C / C ++ під Android. Зазвичай NDK використовується для написання модулів, які потребують високої швидкості виконання. В 2019 році компанію Google поставила мову програмування Kotlin на перше місце в розробці мобільних додатків. Більшість розробників та засновники вже перейшли на цю мову. Тож усі нові пакети для розробки створюються на мові Kotlin.

Архітектура ОС Android поділена на 5 шарів(Рис. 3.3):

1. Ядро Linux(Linux kernel). Ядро виступає в ролі з'єднувача апаратної частини з програмною. ОС Android має доступ до системних служб ядра для керування пам'яттю, роботи з

- драйверами та багатопотоковістю;
2. Середовище виконання(Android Runtime). Android використовує віртуальну машину DALVIK, яка керує запуском додатків в операційній системі. Для кожного додатку створюється окремий процес;
 3. Бібліотеки(Libraries). Даний шар надає розробникам доступ до C / C ++ бібліотек операційної системи ;
 4. Каркас додатків(Application Frsmework). ОС надає загальний доступ до додатків в системі та надає доступ до апаратних можливостей телефону;
 5. Додатки (Applications). Усі додатки, які встановлені на операційну систему.

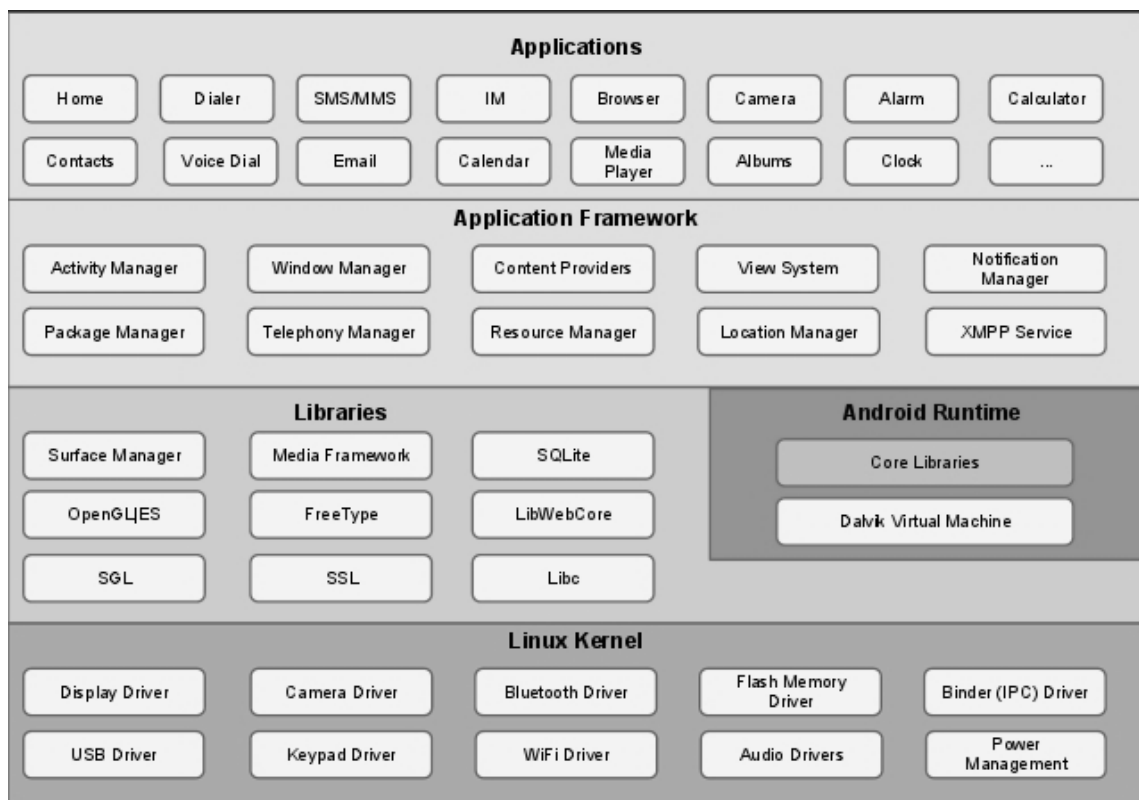


Рис. 3.3 – Архітектура ОС Android[17]

3.1.2 Операційна система IOS

Операційну систему IOS створила компанія Apple. 29 червня 2007 року вона вперше побачила світ на мобільному пристрої iPhone. Дана операційна система є закритою, тому крім компанії засновників її ніхто

більше не може використовувати в ролі операційної системи на мобільних пристроях. Також дана операційна система являється повністю закритою для користувачів. Саме це визначає максимальну турботу про безпеку програмного забезпечення. Apple застосовує всі можливі засоби для забезпечення максимальної безпеки даних своїх користувачів. IOS так само як і Android є Unix подібною ОС. 6 березня 2008 року компанія Apple випустила пакет інструментаріїв для розробників[18].

Архітектура ОС IOS є шарово-орієнтованою. Це означає, що в системі пристуні декілька слоїв з власною зоною відповідальності. Додатки, розроблені для IOS спілкуються з системою за допомогою декількох шарів інтерфейсу. Дана архітектура забезпечує роботу додатків на різних пристроях із різним апаратним складом. Більшість інтерфейсів впроваджено у спеціальних пакетах під назвою фреймворки. Розробники взаємодіють з фреймворками на кожному шару архітектури, щоб використати функціональні можливості системи.

Архітектура ОС IOS складається з наступних шарів:

1. Ядро ОС(Core OS). Відповідає за взаємодію з апаратними частинами системи такими як: Bluetooth, акселератор, зовнішній запам'ятовуючий пристрій, сервіс безпеки, сервіс аутентифікації користувачів;
2. Сервіси ядра(Core Services). Даний шар відповідає за усі сервіси, які надає система користувачу. До них відносяться: адресна книга, синхронізація з хмарними сховищами, сервіс для зберігання даних, тощо;
3. Медіа шар(Media Layer). Забезпечує засоби роботи з аудіо та відео ресурсами. Також відповідає за графічний інтерфейс взаємодії з системою.
4. Cocoa Touch шар. Відповідає за взаємодію системи з користувачем. До нього також входять фреймворки, що працюють з картами, інтеграцією сторонніх сервісів та розробкою інтерфейса користувача.

3.1.3 Мультиплатформна розробка

12 травня 2017 року компанія Google анонсувала фреймворк Flutter, який призначений для мультиплатформної розробки під ОС Android та IOS. Даний фреймворк підтримує мову програмування Dart для створення мультиплатформних додатків. Для ОС Android, Windows, macOS, Linux для роботи додатків використовується віртуальна машина Dart з JIT-компілятором. На відміну від вище сказаних систем, для ОС IOS Flutter використовує AOT-компіляцію. На відміну від інших фреймворків для мультиплатформної розробки, таких як React Native або Xamarin Flutter не використовує нативні віджети операційної системи, на якій запущено додаток. Замість цього він має власний засіб для побудови інтерфейсу, написаний на C/C++. Тому код написаний на Flutter може запросто компілюватися та працювати на різних платформах. Саме тому система комунікацій із зворотнім зв'язком створена на фреймворкі Flutter. Основними перевагами Flutter перед нативною розробкою мобільних додатків є:

1. Одна кодова база працює на різних платформах;
2. Використання власної реалізації рендерингу інтерфейсу додатку;
3. Наявність вбудованого в мову Gabage Collector;
4. Використання AOT та JIT компіляцій;
5. Реалізація декларативного інтерфейса користувача;
6. Наявність бібліотек для уникнення шаблонного коду;
7. Наявність гарячого перезавантаження. Для внесення змін в додаток не потрібно його перекомпілювати;
8. Можливість легко інтегрувати в існуючий проект;
9. Використання об'єктно-орієнтованої мови програмування Dart, яка є новою поміж інших мов, та поєднала найкращі риси багатьох популярних мов програмування;
10. Підвищення швидкості розробки;
11. Наявність прямого доступу до основних функцій системи;
12. Має високу продуктивність відображення інтерфейсу.

					ІАЛЦ.467100.003 ПЗ	Арк.
						42
Зм.	Арк.	№ докум.	Підпис	Дата		

3.2 Компоненти серверної частини

Для реалізації серверної частини було обрано платформу Firebase. Firebase – це платформа-поставник хмарних послуг для клієнтів мобільних та веб додатків. Дана платформа широко використовується у сучасній розробці програмного забезпечення для різних сфер бізнесу. Завдяки сервісам, які надає Firebase, полегшується та пришвидшується розробка мобільних та веб додатків. Дана платформа надає функціонал, який повністю дозволяє позбутися від реалізації власної back-end розробки, тому Firebase є чудовим рішенням для розробників мобільних додатків. Адже інтерфейс платформи максимально простий та надає велику кількість засобів для реалізації клієнт-серверної архітектури. Взаємодія з сервісами платформи відбувається за допомогою інтерфейсів бібліотек Firebase.

Щоб додати Firebase до проекту потрібно встановити плагін FlutterFire та завантажити конфігураційні файли google-services.json для Android та GoogleService-Info.plist для IOS. Завантажуваний файл потрібно перемістити в корінь проекту. Конфігураційний файл містить ключі доступу та адреси сервісів Firebase. Далі плагін за допомогою ключів зможе встановити з'єднання з хмарними сервісами. Всередині бібліотека використовує підключення по сокету для обміну даними між клієнтом та сервером. Для авторизації проекту в системі Firebase використовує OAuth протокол авторизації. Принцип взаємодії клієнта з Firebase зображено на рис. 3.4.

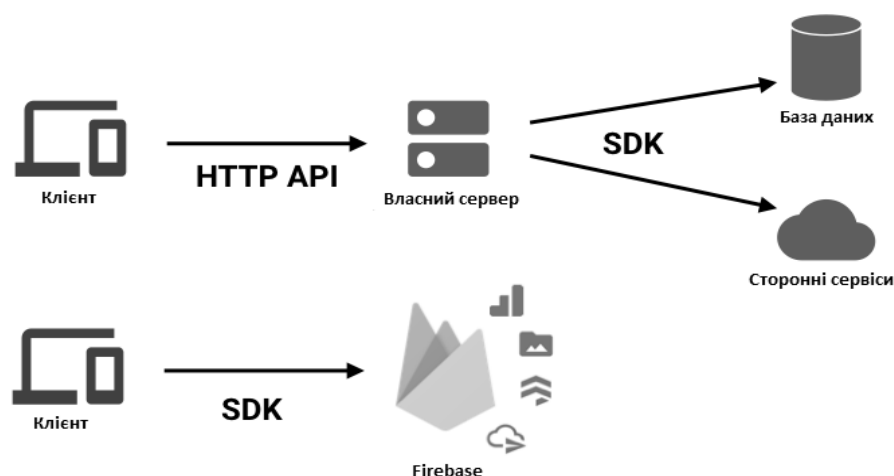


Рис. 3.4 – Різниця взаємодії клієнта з власним сервером та Firebase[21]

Для реалізації системи комунікацій із зворотнім зв'язком були використані наступні сервіси платформи Firebase:

1. Authentication;
2. Firestore;
3. Cloud Functions;
4. Firebase Cloud Messaging;

3.2.1 Сервіс авторизації користувачів

Authentication – сервіс для аутентифікації користувачів, який використовує протокол OAuth 2.0 та OpenId Connect. OAuth 2- це фреймворк для авторизації, що надає додаткам обмежений доступ до призначених для користувача аккаунтів на HTTP сервісах. Він працює за принципом делегування аутентифікації користувача сервісу, на якому знаходиться аккаунт користувача, дозволяючи сторонньому додатку отримувати доступ до аккаунту користувача. OpenID Connect - це рівень ідентифікації, який працює над протоколом OAuth 2.0. Він дозволяє клієнтам перевірити особу користувача на основі аутентифікації, виконаної сервером авторизації, а також отримати основну інформацію про профіль користувача в оперативному режимі та схожий на REST спосіб[22].

Щоб аутентифікувати користувача, система спочатку отримує облікові дані автентифікації від користувача. Ці облікові дані можуть бути електронною адресою та паролем користувача або токеном OAuth від федерального постачальника ідентифікаційних даних. У випадку із системою комунікацій із зворотнім зв'язком, система спочатку дістає OAuth токен у постачальника ідентифікаційних даних Google, а потім передає ці дані в SDK для аутентифікації Firebase. В результаті сервіси Firebase перевіряють валідність цих даних, повертають інформацію облікового запису користувача та створюють окрему сесію для взаємодії з платформой.

Після успішного входу ви можете отримати доступ до основної інформації користувача, а також можете контролювати доступ користувача до даних, що зберігаються в інших продуктах Firebase. Атентифікація за

					ІАЛЦ.467100.003 ПЗ	Арк.
						44
Зм.	Арк.	№ докум.	Підпис	Дата		

допомогою платформи Firebase є зручним способом для інтеграції багатьох інших засобів ідентифікації користувача. Вона працює з такими провайдерами ідентифікаційних даних як: адреса електронної пошти та пароль, оператор мобільного зв'язку, Google, Facebook, Twitter, Github, Yahoo, Microsoft, Apple. Взаємодія платформи з вище перерахованими провайдерами ідентифікаційних даних користувача надає системі можливості для масштабованості та інтеграції у засоби корпоративної взаємодії в компанії. Заміна провайдерів відбувається дуже швидко та лише зі сторони клієнтів.

3.2.2 База даних

Firestore – це документо-орієнтована нереляційна NoSQL база даних від Firebase та Google Cloud Platform. Основною перевагою даного сервісу є синхронізація даних з клієнтами за допомогою слухачів у режимі реального часу. В базу даних влаштоване автоматичне кешування всіх даних, тому робота з нею є максимально зручно, адже синхронізація з серверами відбувається автономно. Робота з Cloud Firestore завжди відбувається через кеш, у разі відсутності інтернету база дані продовжує працювати та синхронізується з сервером при наявності інтернет з'єднання. Cloud Firestore повністю інтегрована з іншими Firebase сервісами, особливо з Cloud Functions. За допомогою Cloud Functions можна налаштувати попередню обробку даних, перед взаємодією з базою даних[23].

Cloud Firestore зберігає дані у вигляді документів, які поєднуються у колекції та підколекції. Кожен документ може містити в собі як просту так і складну структуру об'єктів з полями, що зіставлені із значеннями. Ви також можете створювати підколекції в документах та будувати ієрархічні структури даних, які масштабуються у міру зростання вашої бази даних. Запити до даної бази даних можуть включати в себе широкий набір фільтрів та параметрів сортування. Вони також індексуються за замовчуванням, тому ефективність запитів пропорційна розміру вашого набору результатів, а не набору даних. Інтеграція з сервісів аутентифікації Firebase дозволяє

					ІАЛЦ.467100.003 ПЗ	Арк.
						45
Зм.	Арк.	№ докум.	Підпис	Дата		

захистити дані користувачів за допомогою налаштувань прав доступу. Графічне відображення організації бази даних зображено на рис. 3.5.

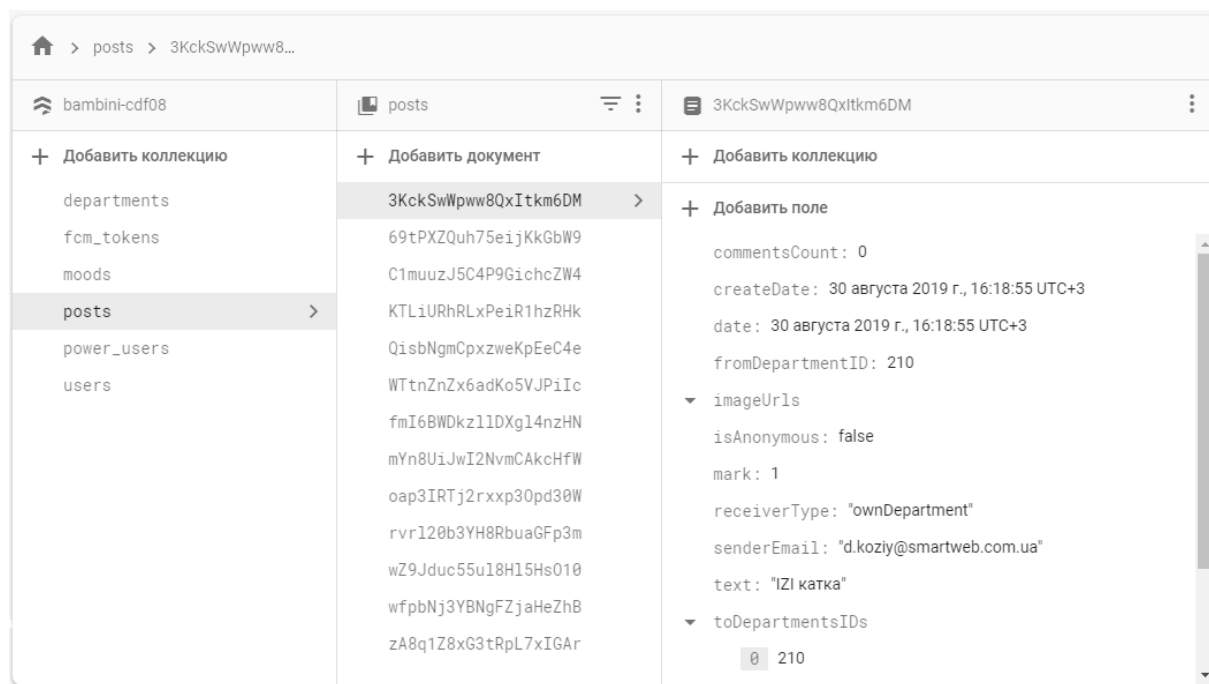


Рис. 3.5 – Організація бази даних системи комунікацій із зворотнім зв'язком

Departments – колекція департаментів компанії. Система в ній зберігає дані про організаційну структуру компанії, коли адміністратор створює простір компанії та заносить туди всю необхідну інформацію. Використовується для створення нового співробітника компанії. Документ колекції має наступні поля:

1. *Id* – унікальний ідентифікатор департаменту;
2. *Name* – назва департаменту;
3. *ParentId* – унікальний ідентифікатор керуючого департаменту над поточним;
4. *HeadId* – унікальний ідентифікатор керівника департаменту;
5. *EmployeeCount* – кількість співробітників в департаменті.

Fcm_tokens – колекція токенів для Firebase Cloud Messaging. Для відправки пуш-повідомлень користувачам, система повинна зберігати токени для зіставлення даних користувача в базі із додатком на телефоні. Документ колекції має наступні поля:

1. *UserId* – унікальний ідентифікатор власника, даного точену;

2. *UserEmail* – електронна адреса користувача;
3. *DepartmentId* – унікальний ідентифікатор департаменту, в якому працює користувач;
4. *UserToken* – токен користувача для сервісу Firebase Cloud Mesdsaging.

Moods – колекція відміток настроїв користувачів. Використовується для зберігання відміток натсрою користувачів, для подальшого аналізу емоційного стану компанії. Документ колекції має наступні поля:

1. *UserId* – унікальний ідентифікатор користувача, який створив відмітку про настрій;
2. *Date* – дата, коли користувач зберіг відмітку;
3. *Mark* – значення настрою користувача;
4. *Comment* – коментар до настрою.

Posts – колекція всіх постів користувачів. Дана колекція зберігає всі пости в системі з відповідними полями для фільтрації та сортування. У кожному документі посту присутня підколекція *Comments*. Документ колекції має наступні поля:

1. *Sender Email* – електрона адреса користувача, який створив пост;
2. *RecieverType* – тип категорії одержувачів посту;
3. *IsAnonymous* – флаг чи даний пост є анонімним;
4. *Mark* – оцінка зворотного зв'язку;
5. *Text* – текст зворотного зв'язку;
6. *ToDepartmentsIds* – масив унікальних ідентифікаторів департаментів, які повинні отримати пост;
7. *ToUsersIds* – масив унікальних ідентифікаторів керівників, які повинні отримати пост;
8. *UsersLikesIds* – масив унікальних ідентифікаторів користувачів, які оцінили пост;
9. *CommentsCount* –кількість коментарів до посту;
10. *Date* – дата створення посту;

					ІАЛЦ.467100.003 ПЗ	Арк.
						47
Зм.	Арк.	№ докум.	Підпис	Дата		

11. *FromDepartmentId* – унікальний ідентифікатор департаменту, в якому працює автор посту.

Документ підколекції *Comments* має наступні поля:

1. *SenderEmail* – електронна адреса автора коментарю;
2. *Text* – текст коментарю;
3. *Date* – дата створення коментарю;
4. *UserLikesIds* – масив унікальних ідентифікаторів користувачів, які оцінили коментар.

Power_users – колекція адміністраторів системи. Дана колекція зберігає посилання на користувачів системи, які мають права адміністратора. Документ колекції має наступні поля:

1. *Id* – унікальний ідентифікатор адміністратора;
2. *UserId* – унікальний ідентифікатор об'єкта користувача.

Users – колекція користувачів системи. Документ колекції має наступні поля:

1. *Id* – унікальний ідентифікатор користувача;
2. *FirstName* – ім'я користувача;
3. *LastName* – прізвище користувача;
4. *IsHead* – флаг, який вказує чи є користувач керівником відділу;
5. *Email* – електронна адреса користувача;
6. *DepartmentsIds* – масив унікальних ідентифікаторів департаментів, в яких працює користувач.

3.2.3 Хмарні функції

Cloud Functions - це безсерверний фреймворк, який дозволяє автоматично запускати код при створенні подій, викликаних функціями Firebase та HTTPS-запитами до платформи. Код хмарних функцій, написаних на JavaScript або TypeScript зберігається в хмарі Google. Для роботи функцій, не потрібно реалізовувати та керувати власними серверами. Розробник пише

					ІАЛЦ.467100.003 ПЗ	Арк.
						48
Зм.	Арк.	№ докум.	Підпис	Дата		

функції та потім розгортає їх на серверах Google, які в свою чергу починають негайно керувати функцією. Функцію можна запустити двома способами: за допомогою HTTP-запиту, або, у випадку фонових функцій, сервери Google підписуються й слухають події, під час яких спрацьовує функція, та запускають функцію під час виникнення події[24].

Кожна функція ізольована та має власне середовище та конфігурацію для роботи. У випадку, якщо навантаження збільшується чи зменшується, сервера Google швидко змінюють кількість ресурсів віртуального сервера, необхідних для роботи функції.

Для створення хмарних функцій потрібно написати код функції, обравши при цьому постачальника події (наприклад, база даних Firestore) та вказуючи умови, при яких функція запускається. Далі цю функцію потрібно розгорнути на серверах Firebase, який в свою чергу самостійно підключає функцію до обраного постачальника подій. Під час створення події постачальником, умови якої відповідають критеріям запуску функції, викликається код. Кожен раз, після оновлення коду функції, очищаються старі екземпляри та замінюються новими. При видаленні функції всі екземпляри очищаються та видаляється з'єднання між функцією та провайдером подій.

Щоб реалізувати функціонал для обробки загальних даних у синхронному для всіх клієнтів режиму, були створені функції збільшення та зменшення коментарів до посту. Так як коментарі можуть додаватися одночасно, їх кількість, що відображається у деталях до посту, втрачає актуальність, тому було прийнято рішення написати функції, які б викликалися при події створення або видалення коментаря в під колекції *Comments*. Таким чином кількість коментарів буде оновлюватися в одному місці та відсилати усіх клієнтів про нове значення. Також в системі комунікацій із зворотнім зв'язком написані хмарні функції для розсилки пуш-повідомлень користувачам при додаванні нових постів або коментарів. Дана функція викликається, коли створюється відповідний документ в базі даних.

					ІАЛЦ.467100.003 ПЗ	Арк.
						49
Зм.	Арк.	№ докум.	Підпис	Дата		

Основний її функціонал – це перехопити створений документ, визначити множину користувачів, які мають доступ до перегляду даного посту чи коментарю, дістати з бази даних токени Google Cloud Messaging для ідентифікації клієнтів та розіслати пуш-повідомлення по цим токенам. Реалізація попередньо описаної функції зображена на рис. 3.6.

```
exports.triggerGeneralNotifications = functions.firestore.document('/posts/{documentId}')
.onCreate(async (snap, context) => {
  const postRef = firestore.doc(`posts/${context.params.documentId}`);
  let post = (await postRef.get()).data();
  post.id = postRef.id
  let receiverType = post.receiverType
  switch(receiverType) {
    case "evo":
      sendGeneralNotifications(post)
      break;
    case "betweenDepartments":
      sendInteractionNotifications(post)
      break;
    case "ownDepartment":
      sendOwnDepartmentNotification(post)
      break;
  }
});
```

Рис. 3.6 – Реалізація Cloud Function для розсилки пуш-сповіщень

3.2.4 Сервіс пуш-сповіщень

Firebase Cloud Messaging - це рішення для обміну повідомлень між платформами. Використовуючи FCM, можна відсилати повідомлення на пристрої клієнтів без витрат ресурсів сервера. Кожен мобільний додаток використовує даний сервіс для централізованого керування пуш-сповіщеннями. Максимальний розмір повідомлень не повинен перевищувати 4 КБ. Але цього достатньо для відображення інформації користувачу у вигляді сповіщення, що спонукає його на наступну роботу з додатком[25].

Реалізація FCM вимагає два компоненти для відправлення та отримання повідомлень: довірене середовище, наприклад хмарні функції для Firebase або власний сервер, та клієнтська частина (iOS, Android або веб), яка отримує повідомлення за допомогою сокетів.

В системі комунікацій усі пуш-сповіщення про нові пости та коментарі приходять з сервісів Firebase за допомогою Google Cloud Messaging. Хмарні

функції встановлені на відповідні тригери до хмарної бази даних та відправляють пуш-повідомлення на сторону клієнта. Даний підхід вимагає мінімальних затрат ресурсів для реалізації поставлених завдань. За допомогою Google Cloud Messaging можна відправляти таргетингові повідомлення з вибіркою клієнтів, таким чином система набуває гнучкості у роботі з користувачами та втримує своїх користувачів, нагадуючи про своє існування.

3.3 Архітектура клієнтської частини

Для реалізації клієнтської частини було використано фреймворк Flutter для мобільних пристроїв. Для написання масштабованого, зрозумілого та тестованого коду було використано чисту архітектуру для взаємодії модулів проекту та принципи SOLID для уникнення критичних ділянок та помилок. Принцип чистої архітектури широко використовується у побудові додатків для мобільних пристроїв. Вони набрали своєї популярності за допомогою простих правил, які роблять код більш читабельним та легким до змін. Основна ідея чистої архітектури полягає в поділенні модулів проекту на шари з власною відповідальністю. Щоб уникнути високого рівня залежностей між цими шарами, згідно з принципами SOLID, взаємодія між ними повинна відбуватися виключно за допомогою інтерфейсів. В такому разі кожен компонент системи стає легко підмінюваним. Основними шарами чистої архітектури є: шар представлення, шар бізнес-логіки, шар даних.

Шар представлення(Presentation layer) працює на рівні роботи з клієнтом. Основною його задачею є обробка взаємодії користувача з додатком, перенаправлення потоку даних в інший шар та отримання оброблених даних і їх відображення. Даний шар взаємодіє лише з бізнес-логікою проекту. Для організації коду на даному рівні в мобільних додатках використовуються декілька патернів: MVP, MVVM, MVI. В системі комунікацій було використано патерн MVVM(Рис. 3.7). MVVM розшифровується як Model-View-ViewModel. Model в даному випадку поєднує в собі бізнес-логіку та шар даних[26]. View – це представлення інтерфейса користувача. ViewModel – це проміжна ланка, яка слугує для

отримання подій користувача, їх обробка та передача на наступний шар або отримання даних з Model та адаптація їх до вигляду інтерфейсу. На даному етапі компонент Model являє собою чорний ящик, адже він працює з обробкою, відправленням та завантаженням усіх даних, з якими працює користувач. Основний акцент в шарі представлення падає на взаємодію View, ViewModel та Model.

Основний принцип роботи даного патерна полягає в Observer/Subscriber взаємодії компонентів. View має в собі посилання на ViewModel, таким чином воно завжди інформує ViewModel про будь-які дії користувача. Основною відповідальністю View є передача інформацію про дії користувача далі та відобразити дані, які оновлюються у ViewModel. В свою чергу ViewModel є незалежною від View, тому що вона не має посилання на нього. Замість цього вона має інтерфейс потоків даних, які оновлюються залежно від стану Model. Кожен раз при взаємодії користувача з додатком, ViewModel отримує інформацію від View про події, потім ViewModel починає оновлювати стан Model. Після обробки всіх даних Model повертає оброблені дані. ViewModel отримує нові данні та змінює свій стан, в цей час View, яка прослуховує зміни стану ViewModel, отримує нову порцію даних для відображення.

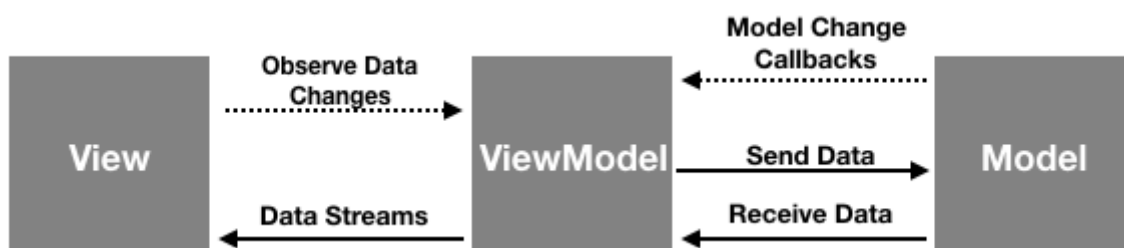


Рис. 3.7 – Принцип роботи патерна MVVM[26]

Наступним шаром у архітектурі проекту є бізнес-логіка(Domain Layer). Дана частина описує об'єкти реального світу(Entity), які перенесені в програму, та методи обробки цих об'єктів(Use cases). Основною особливістю даного шару є те, що він не повинен мати залежностей на інші компоненти, бібліотеки чи фреймворки. Вся логіка повинна бути написана лише засобами мови програмування. Це надасть можливість майбутнього пере використання

даної програмної частини. Основною задачею шару бізнес-логіки, є отримання команд та даних для обробки від шару представлення. Залежно від типу функціоналу виконуються перетворення з об'єктами. Вже оброблені об'єкти передаються далі для збереження або відправлення на сервер. Саме тут розташована основна логіка додатку.

Наступним і не менш важливим компонентом є шар даних (Data Layer). Для іншої частини програми він являє собою інтерфейс для отримання або збереження даних. Для користувачів шару даних не важливо звідки він бере ці дані та куди зберігає їх. Але всередині потрібно чітко організувати принцип збереження даних. Саме в цьому місці додаток синхронізується з серверами, зберігає дані в кеш-пам'ять та вирішує, який спосіб збереження або завантаження даних використовувати. Для даного шару в мобільних додатках використовується патерн під назвою Repository (Рис. 3.8). Це патерн, який надає всім іншим компонентам простий інтерфейс для збереження та отримання даних. Але в своїй реалізації він поєднує роботу як із зовнішніми, так із внутрішніми способами збереження даних.

Основною задачею шару даних є перетворення об'єктів бізнес-логіки у вигляд, який має база даних та їх збереження. Вся робота з мережею відбувається тут. У випадку з системою комунікацій та зворотнім зв'язком, даний шар працює з базою даних Firestore, за допомогою Firebase SDK. В цьому місці в додатку реалізовано кешування всіх даних, тож система може працювати в оффлайн режимі при відсутності з'єднання з інтернетом та синхронізувати дані при отриманні доступу до мережі. Для відділення роботи з хмарною базою даних та локальною базою даних використовуються об'єкти DataSource, які вміють працювати зі своєю базою даних. Для збереження даних локально було використано базу даних SQLite. SQLite – це реляційна локальна база даних, яка має програмні інтерфейси для взаємодії з мобільними додатками. Основний потік роботи з даними на даному шарі наступний: репозиторій отримує дані від шару бізнес-логіки для їх подальшого збереження або відправлення на сервер. Отримані дані трансформуються у вигляд, зрозумілий для бази даних, а потім за допомогою відділеного ресурсу даних синхронізуються з сервісами Firebase. Після збереження та

обробки даних користувача в хмарній базі даних, повертаються результати на мобільний девайс. Репозиторій, за допомогою локального ресурсу даних зберігає оброблені результати та трансформує їх у вигляд, зрозумілий для бізнес-логіки. Після всіх цих кроків оброблені або отримані з Firestore дані передаються до шару бізнес-логіки для подальшого їх використання системою.

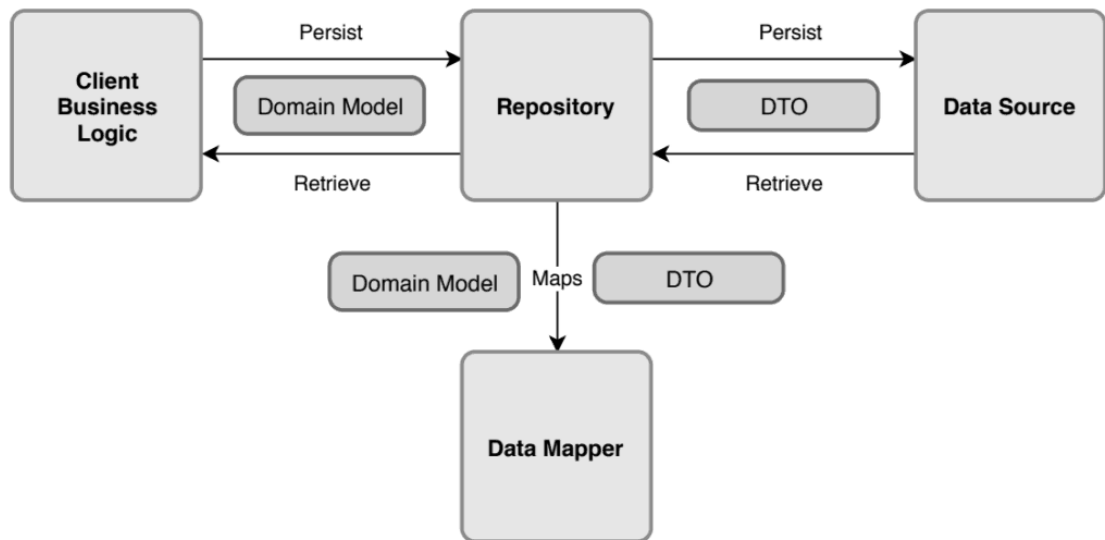


Рис. 3.8 – Принцип роботи патерна Repository[27]

Поєднання вище описаних архітектурних підходів дозволяє нам реалізувати принцип чистої архітектури в мобільному додатку. За допомогою наведених патернів та використання інтерфейсів взаємодії між компонентами, усі частини системи є легко замінювані.

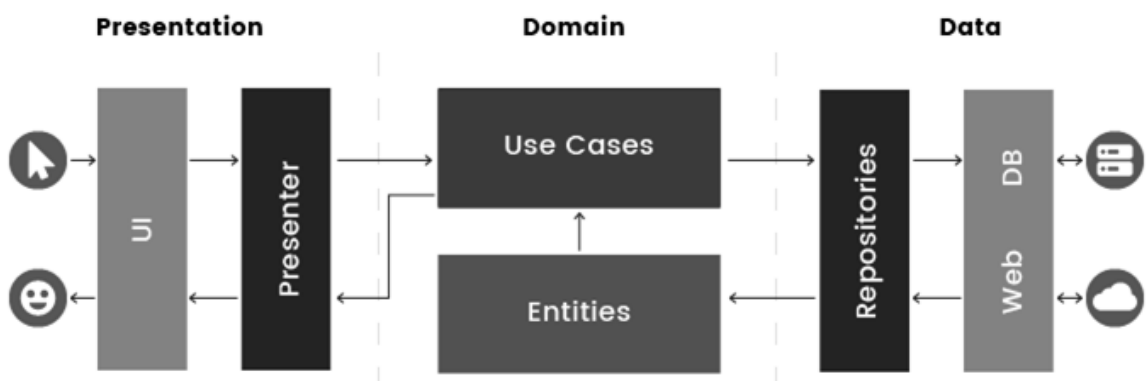


Рис. 3.8 – Схема взаємодії компонентів за принципами чистої архітектури

3.4 Компоненти клієнтської частини

Проект складається з 6 модулів, кожен з яких представляє окремий та

незалежний від інших екран та його функціонал. Кожний модуль побудований за принципами чистої архітектури із реалізацією описаних вище патернів проектування. На рис. 3.8 зображено структуру пакетів проекту.

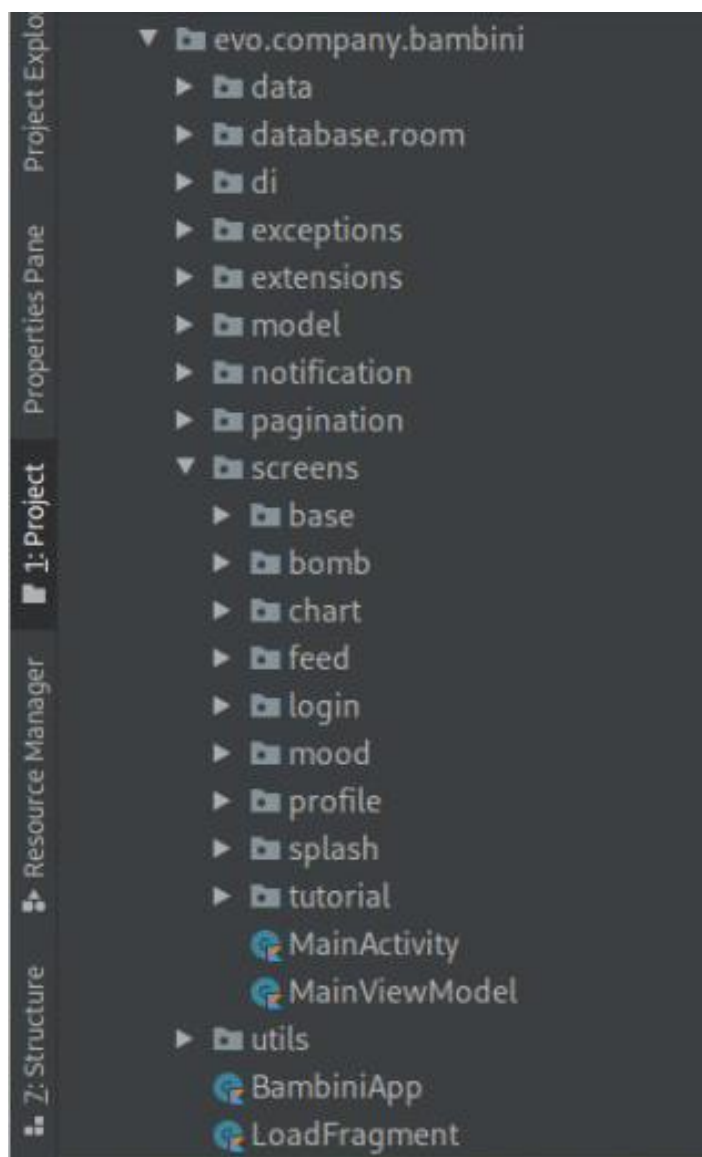


Рис. 3.9 – Структура пакетів проекту

data – пакет, який відповідає за шар даних, у ньому знаходяться всі класи, що відповідають за роботу з хмарною базою даних Firestore та локальною базою даних. Основна робота з хмарними сервісами та збереженням даних знаходиться тут. В проекті реалізовано наступні репозиторії, кожен з яких працює з власним набором класів DataSource:

1. *ProfileRepository* – відповідає за збереження й оновлення корпоративних даних користувача та синхронізацію налаштувань роботи пуш-повідомлень з сервісами Firebase;

2. *LoginUserRespository* – відповідає за збереження та синхронізацію ідентифікаційних даних користувача. Має набір маперів для перетворення записів з бази даних в об'єкти бізнес логіки;
3. *DepartmentRespository* – відповідає за синхронізацію департаментів компанії з хмарною базою даних Firestore. Також мапить та поєднує записи департаментів та керівників департаментів з таблиці Users;
4. *PostRepository* – відповідає за роботу з постами. Реалізує широкий набір методів для створення, збереження, оновлення, видалення та синхронізацію в реальному часі постів. Включає в себе набір маперів для трансформації об'єкта Post;
5. *MoodRepository* – відповідає за збереження відміток про настрій користувача та завантаження оброблених для аналізу даних емоційного стану компанії та окремого користувача також;
6. *FCMRepository* – відповідає за збереження та видалення FCM токенів, які потрібні для відправки пуш-повідомлень користувачам;
7. *CommentRepository* – відповідає за збереження та синхронізацію коментарів до постів;
8. *PreferencesHelper* – відповідає за збереження локальних налаштувань користувача.

database.room – пакет, який містить в собі конфігураційні класи локальної бази даних та інтерфейси DAO для роботи з цією базою;

di – пакет, який містить компоненти провадження залежностей класів. Даний пакет реалізує модуль, який керує всіма залежностями в проекті, створює екземпляри класів та ініціалізує об'єкти з потрібними для них залежностями;

exceptions – включає класи обробки критичних помилок, які виникають під час роботи додатку;

extensions – пакет з файлами, які складаються з функцій розширень до класів у проекті;

					ІАЛЦ.467100.003 ПЗ	Арк.
						56
Зм.	Арк.	№ докум.	Підпис	Дата		

model – пакет шару бізнес логіки. Він включає в себе класи даних, з якими працює програма. На рис. 3.10 зображено діаграму відношень між класами даних бізнес логіки системи.

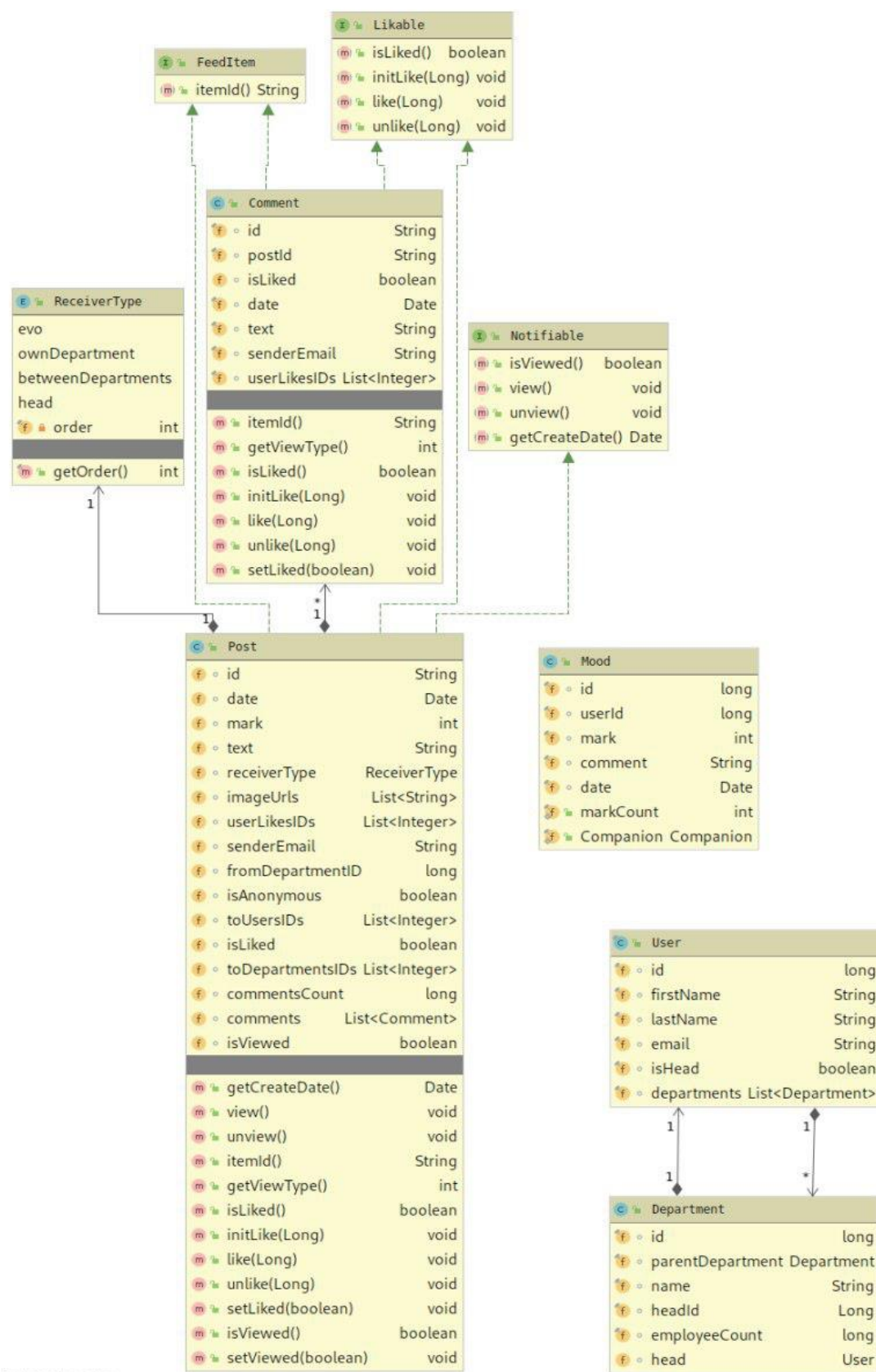


Рис. 3.10 – Діаграма класів бізнес-моделей

notification – пакет, який містить класи для роботи з отриманням та відображенням пуш-сповіщень із сервісу FCM.

pagination – пакет, який містить сукупність класів для реалізації пагінації. Тобто за допомогою даного пакету додаток завантжає всі дані порціями по мірі потреби користувача. Даний підхід допомагає значно заощадити ресурси системи та трафіка.

screens – пакет містить всі класи групувані відповідно до екрану, функціоналу, якого вони реалізують. Саме в цьому місті розташовані класи інтерфейсу користувача та реалізація всієї логіки додатку.

Utils – даний пакет включає в себе файли з програмними утилітами.

					ІАЛЦ.467100.003 ПЗ	Арк.
						58
Зм.	Арк.	№ докум.	Підпис	Дата		

Висновки до розділу 3

У даному розділі було детально описано платформи для реалізації серверної та клієнтської частини системи комунікацій із зворотнім зв'язком та їх компоненти, які були використані для реалізації поставлених завдань. З боку клієнта було обрано мультиплатформенний фреймворк Flutter. У якості серверної частини було взято платформу Firebase, адже вона дає широкий та гнучкий інтрусентарій для повної реалізації функціоналу системи. Система використовує наступні сервіси Firebase:

- Firebase authentication;
- Firestore;
- Firebase Cloud Messaging;
- Firebase Cloud Functions;

Також було описано основні архітектурні підходи у реалізації клієнтської частини та детальне обґрунтування причин використання кожного підходу та патерну. У розділі описується структура та порядок роботи клієнтської частини системи, адже основний функціонал розташований саме в ній.

					ІАЛЦ.467100.003 ПЗ	Арк.
						59
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 4

ІСТРУКЦІЯ КОРИСТУВАЧА

В даному розділі детально описано інтерфейс користувача системи комунікацій із зворотнім зв'язком. Також наведені пояснення по використанню кожного із екрану, їх функціонал та можливі способи взаємодії. Щоб понуритися в комунікаційний процес та почату активну роботу в ньому, для початку необхідно просто завантажити мобільний додаток на ваш телефон. Система підтримує ОС Android та ОС IOS.

Після того, як ви завантажите додаток з Google Play Market або App Store, необхідно запустити додаток та натиснути кнопку «Sign in» (Рис. 4.1).

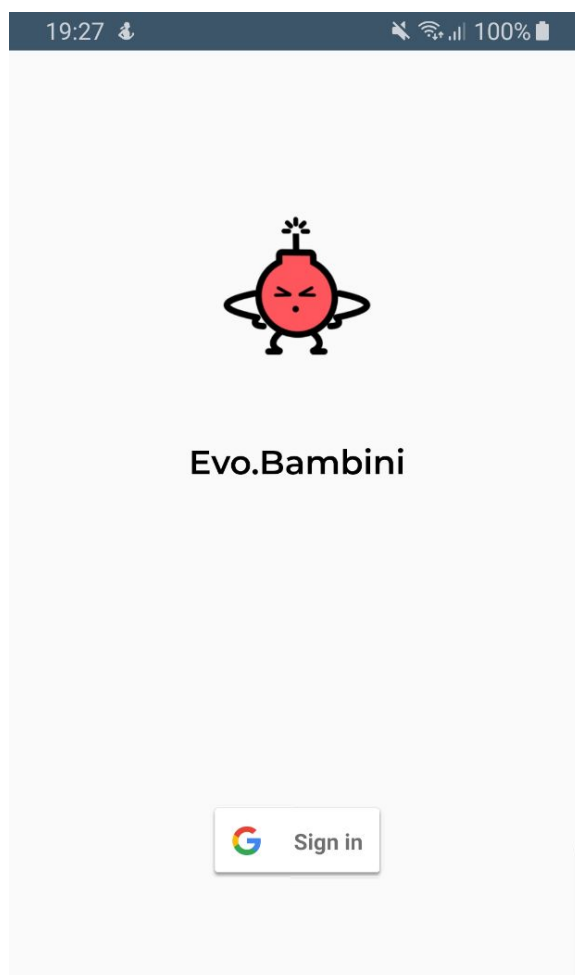


Рис. 4.1 – Екран входу в додаток

Аутентифікація відбувається за допомогою Google акаунту, тож система відкриє список авторизованих на мобільному пристрої Google акаунтів(Рис. 4.2). Користувачу потрібно обрати акаунт для авторизації.

					ІАЛЦ.467100.003 ПЗ	Арк.
						60
Зм.	Арк.	№ докум.	Підпис	Дата		

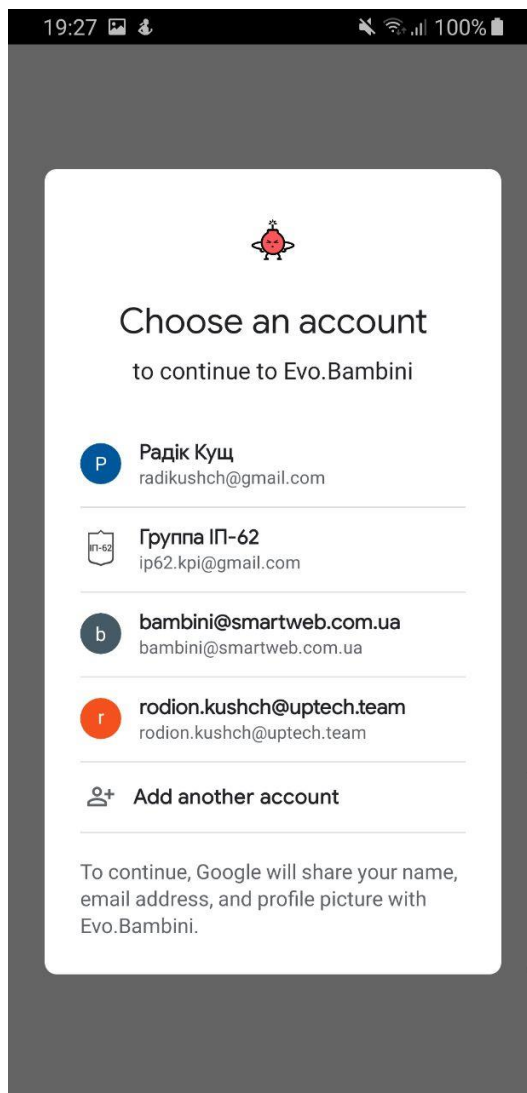


Рис. 4.2 – Екран вибору аккаунту для авторизації

Після успішної авторизації додаток пере направить користувача на головний екран, який має 4 вкладки та кнопку «Профіль» у верхньому правому куті екрану. Автоматично відкривається екран відправлення зворотнього зв'язку під назвою «Бомба». На ланому екрані користувач має можливість почати процедуру створення зворотного зв'язку. Залишити зворотній зв'язок дуже просто. Для цього необхідно просто свайпнути кнопку бомби у пітрібний бік (рис. 4.3):

1. Проведіть пальцем в лівий бок по екрану, якщо вас щось не влаштовує на даний момент і у вас поганий настрій.

2.Проведіть пальцем по екрану в правий бік, якщо ви відчуваєте себе в цілковитій гармонії.

					ІАЛЦ.467100.003 ПЗ	Арк.
						61
Зм.	Арк.	№ докум.	Підпис	Дата		



Рис. 4.3 – Бомба зворотнього зв'язку

Після чого система відкриє екран з формою фідбеку(Рис. 4.4), щоб ви більш детально описали ситуацію, що сталося або, що конкретно вас турбує та не влаштовує. На цьому етапі ви також визначаєте кому має надійти даний зворотній зв'язок:

- Керівництву
- Вашому відділу
- Іншому відділу
- На всю компанію.

А також на цьому кроці перед вами стоятиме вибір, залишити відгук анонімно або ж вказати своє ім'я. Натиснувши на кнопку «Надіслати» система відправить пост, розішле пуш-сповіщення всім одержувачам та відкриє головний екран з бомбою.

					ІАЛЦ.467100.003 ПЗ	Арк.
						62
Зм.	Арк.	№ докум.	Підпис	Дата		

22:17

13%

×

Расскажи подробнее

Не стесняйся, говори своими словами

Укажи, кого касается твой отзыв

Evo

Указывать имя

Поделиться

Рис. 4.4 – Экран відправлення зворотного зв'язку

У випадку, якщо користувач обере категорію одержувачів посту «Інший відділ», система відкриє екран пошуку потрібного відділу (Рис. 4.5). На якому можна знайти потрібний вам відділ, увівши в рядку пошуку або ім'я та прізвища керівника або назву розділу. По мірі вводу символів система буде знаходити схожі результати та фільтрувати список відділів.

22:17

13%

×

Название отдела

Поиск по отделам и руководителям

EVO.company

Николай Палиенко

Crafta

Ольга Шлапунова

Команда контента

Алина Карабут

Команда поискового продвижения сайтов (SEO)

Артем Маруженко

Команда размещения PPC рекламы для клиентов Prom+

Владислав Ляховский

Рис. 4.5 – Экран пошуку відділу

З головного екрану користувач може перейти на екран «Профіль» (Рис.4.6), натиснувши кнопку в лівому верхньому куті. Система відкриє екран з інформацією про співробітника та списком налаштувань push-сповіщень. На екрані можна вимкнути звук на наступні дії:

- Коментар до посту;
- Нові записи відділу;
- Нові записи компанії;

Час сповіщення про відмітку емоційного стану. На даному екрані також присутня форма для зворотнього зв'язку розробникам, яка створена з метою врахування всіх побажань стосовно роботи даного додатку в майбутньому, та кнопка виходу з акаунту.

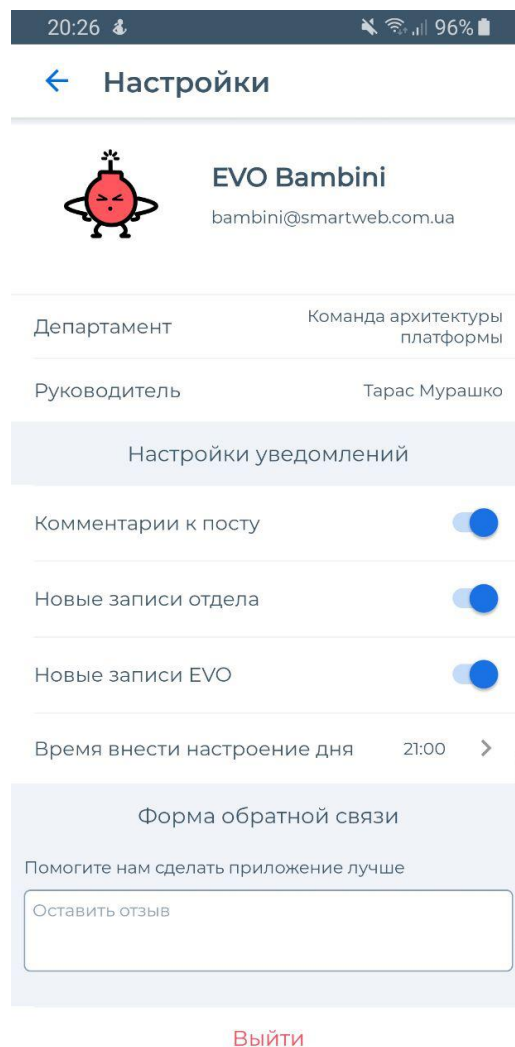


Рис. 4.6 – Экран наладки профиля

На вкладці «Настрій» користувач може зберігати свій настрій в системі для подальшого його аналізу(Рис. 4.7). В подальшому буде можливість порівняти цикли настрою. Для відмітки настрою, потрібно обрати ваш теперішній настрій з 5 запропонованих варіантів. Після виходу з цього екрану додаток збереже останній обраний користувачем настрій. Наступного разу, при вході на екран, обраний варіант буде виділено та зображено зверху над усіма іншими.



Рис. 4.7 – Экран выбора настрою

Зайшовши у вкладку «Графік»(Рис. 4.8) ви побачите проаналізовані динаміку стану вашого настрою. На даному екрані відобрежний графік середнього натсрою всіх співробітників в компанії за останній тиждень та діаграму відношень показників настрою користувача за останній місяць. Даний аналіз допомагає співробітникам та керівникам слідкувати за

					ІАЛЦ.467100.003 ПЗ	Арк.
						65
Зм.	Арк.	№ докум.	Підпис	Дата		

станами емоційного задоволення роботи в компанії й відповідно робити
ВИСНОВКИ.



Рис. 4.8 – Аналітика настрою за певний проміжок часу

Увійшовши на вкладку «Обговорення»(Рис. 4.9), користувач отримує доступ до списку із постами зворотного зв'язку, поділеними на категорії. Біля кожного посту відображено кількість користувачів, які його оцінили, та кількість коментарів до посту. Щоб оцінити пост можна клікнути на кнопку «Палець вгору» або два рази клікнути на пост. Свайпом вліво або вправо по списку, буде віображена нова категорія постів зі своїм списком. При наявності нових непрочитаних постів, вони будуть помічені красною крапкою в верхньому правому куті. При про листуванні або відкритті посту, він автоматично стає прочитаним для користувача.

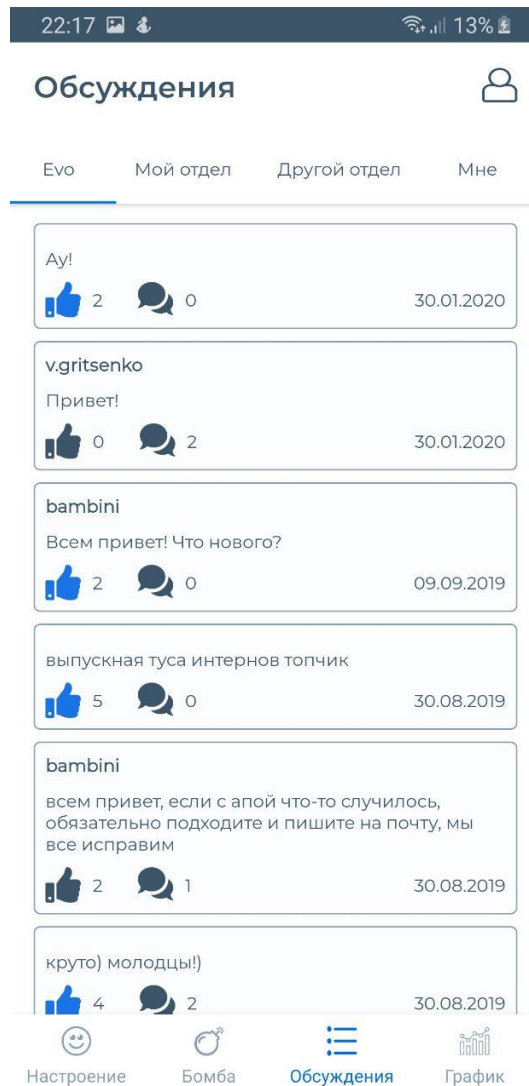


Рис. 4.9 – Экран обговорення зворотного зв'язку

Для перегляду детальної інформації посту та списку коментарів, потрібно натиснути на пост із списку. Система відкриє екран з деталями до посту (Рис. 4.10). На даному екрані користувач може оцінити пост або коментар, клікнувши два рази на нього або на кнопку «Палець вгору». Також користувач може надіслати коментар до посту, увівши текст у відповідній формі та натиснувши кнопку «Відправити». Система миттєво відобразить новий коментар в списку у всіх користувачів.

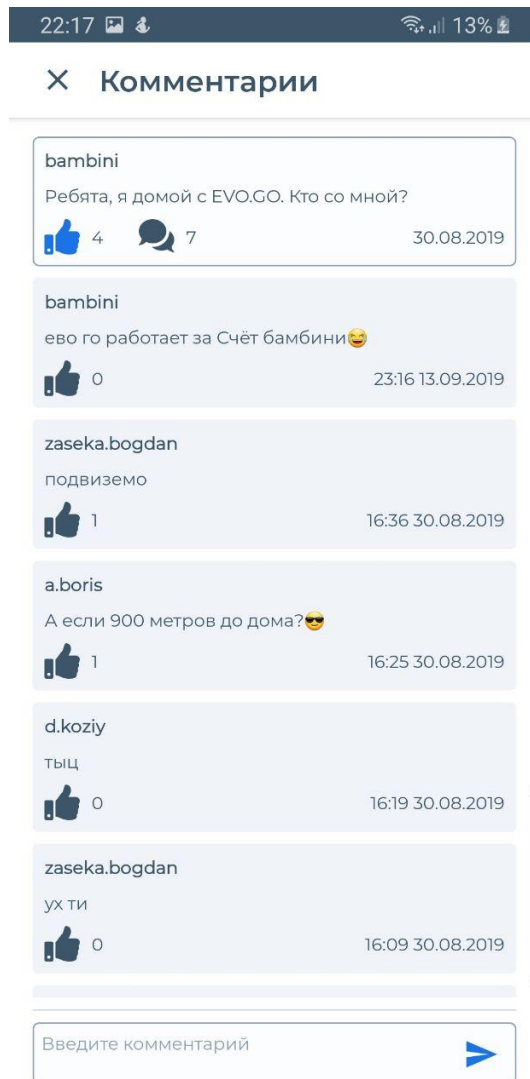


Рис. 4.10 – Экран с деталями до посту

При отримання нових постів або коментарів, додаток повідомляє про зміни за допомогою пуш-сповіщень та відповідного звукового сигналу. Натиснувши на сповіщення, відривається екран з деталями посту.

Висновки до розділу 4

У даному розділі було детально описано інтерфейс користувача. Було наведено скриншоти кожного з екрану в додатку та описано увесь функціонал кожного екрану. За допомогою скриншотів було написано детальну інструкцію, як працює програма з прикладами. Реалізація системи основана на створенні окремих і незалежних між собою екранів, кожен з яких має свою зону відповідальності. Це дозволяє з легкістю додавати нові екрани до існуючого функціоналу або видаляти старі. У разі вдосконалення функціоналу, користувач матиме повніст. Робочий функціонал попередньої версії додатку, адже компоненти незалежні один від одного.

Створений інтерфейс системи інтуїтивно зрозумілий, простий та приємний на вигляд. Даний дизайн дозволяє користувачам з легкістю почати користуватися системою та насолоджуватися її зовнішнім виглядом.

					ІАЛЦ.467100.003 ПЗ	Арк.
						69
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У даній роботі було досліджено тему ефективності впровадження комунікації та зворотнього зв'язку в робочий процес компанії. Було виконано огляд популярних існуючих рішень поставлених завдань, та виявлено, що не існує системи, яка базується на створення процесу комунікації навколо зворотного зв'язку. Тому перед нами було поставлене завдання створити таку систему. Для реалізації було проведено дослідження та пошук вимог до даного програмного забезпечення. Відповідно до посоставлених вимог було спроектовано та розроблено систему з відповідним функціоналом, яким дозволяє вирішити поставлені проблеми. Процес дослідження та його результати були детально описані в даній дипломній роботі.

В результаті була створена система комунікацій із зворотнім зв'язком для корпоративного використання, яка працює на базі мобільних операційних систем Android та IOS. Дана система спроектована згідно з сучасними способами розробки програмного забезпечення для мобільних пристроїв. Програмне забезпечення може використовуватися будь-якою компанією для досягнення поставлених цілей, а саме: вдосконалення процесу комунікації між співробітниками, впровадження автоматизованого та зручного рішення для поширення зауважень та пропозиції співробітників, а також можливість відслідковування емоційного стану співробітників компанії та його аналізу. Дана система є зручною для розуміння та використання, тому користувачам не знадобиться додаткових зусиль для освоєння роботи в ній.

					ІАЛЦ.467100.003 ПЗ	Арк.
						70
Зм.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ:

1. Єжи Меллібруда Я-ти-ми Психологічні можливості покращення міжособистісних контактів Варшава: Nasz Księgarnia, 1980 М .: Прогресс, 1986
2. КОМУНІКАЦІЙНИЙ ПРОЦЕС В УПРАВЛІННІ:ОСНОВНІ ПОЛОЖЕННЯ. О. І. Бабчинська [Електронний ресурс] – режим доступу: http://www.economy.nayka.com.ua/pdf/9_2018/51.pdf
3. Структура комунікативного процесу [Електронний ресурс] – режим доступу: <http://studies.in.ua/ru/sociologiya-seminary/2499-struktura-komunkativnogo-procesu.html>
4. 14 правил фідбеку [Електронний ресурс] – режим доступу: <https://happymonday.ua/ru/gramotnyj-fidbjek-14-pravil>
5. Схема зворотнього зв'язку [Електронний ресурс] – режим доступу: https://uk.wikipedia.org/wiki/%D0%97%D0%B2%D0%BE%D1%80%D0%BE%D1%82%D0%BD%D0%B8%D0%B9_%D0%B7%D0%B2%27%D1%8F%D0%B7%D0%BE%D0%BA#/media/%D0%A4%D0%B0%D0%B9%D0%BB:Ideal_feedback_model.svg
6. Міжособистісний зв'язок та його різновиди [Електронний ресурс] – режим доступу: https://pidruchniki.com/90384/sotsiologiya/mizhosobistisniy_zvorotniy_zvyazok_riznovid_i_konstruktivniy_destruktivniy
7. Дослідження Гарвардської школи бізнесу стосовно користі фідбеку [Електронний ресурс] – режим доступу: <https://hbr.org/2020/06/good-feedback-is-a-two-way-conversation>
8. Slack [Електронний ресурс] – режим доступу: <https://slack.com/intl/en-ua/>
9. Slack переваги та недоліки використання [Електронний ресурс] – режим доступу: <http://slaidik.com.ua/slack-oglyad-mesendzhera-dlya-produktivnoyi-spilnoyi-roboti/>
10. Інтерфейс десктопної версії Slack [Електронний ресурс] – режим доступу: <https://telegraf.design/news/slack-onovyv-veb-versiyu-ta-desktop-kliiyent/>

					ІАЛЦ.467100.003 ПЗ	Арк.
						71
Зм.	Арк.	№ докум.	Підпис	Дата		

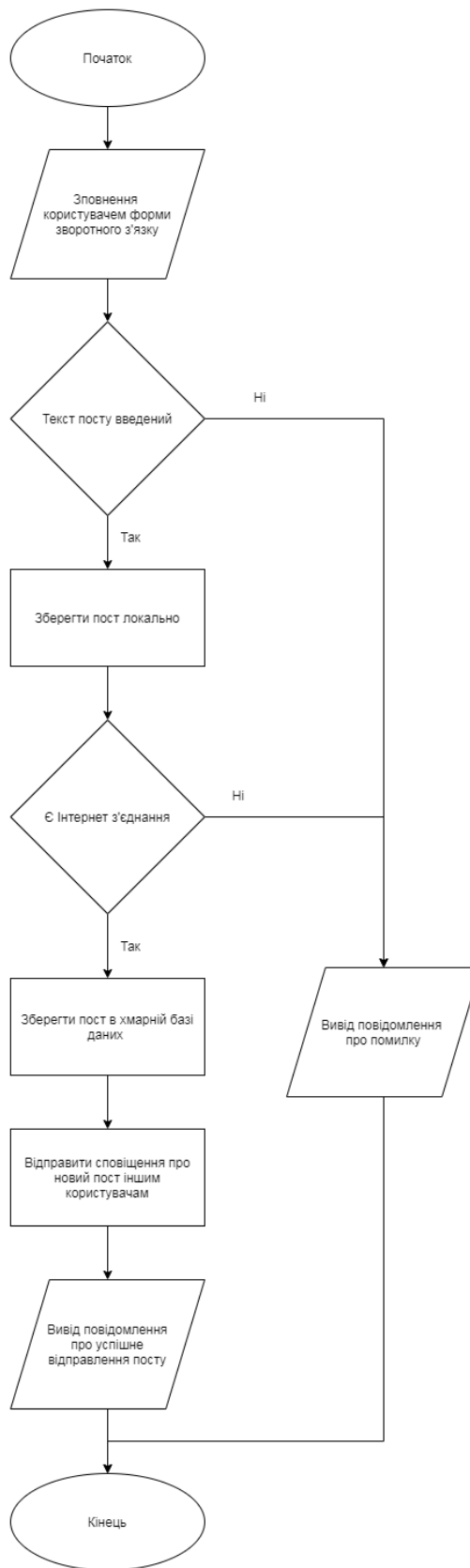
11. Google Forms [Електронний ресурс] – режим доступу:
https://www.google.com/intl/ru_ua/forms/about/
12. Plai [Електронний ресурс] – режим доступу: <https://plai.team/>
13. Happy Job [Електронний ресурс] – режим доступу: <https://happy-job.ru/>
14. Частка використання мобільних операційних систем
<https://netmarketshare.com/>
15. Операційна система Android [Електронний ресурс] – режим доступу:
<http://ipkey.com.ua/faq/912-android.html>
16. Версії операційної системи Android <https://captain-droid.com/uk/android/>
17. Архітектура ОС Android <https://studfile.net/preview/3651536/page:2/>
18. Операційна система IOS [Електронний ресурс] – режим доступу:
<http://ipkey.com.ua/faq/935-apple-ios.html>
19. Архітектура ОС IOS [Електронний ресурс] – режим доступу:
<https://studbooks.net/2249813/informatika/arhitektura>
20. Мультиплатформна розробка [Електронний ресурс] – режим доступу:
<https://uk.wikipedia.org/wiki/Java>
21. Платформу Firebase [Електронний ресурс] – режим доступу:
<https://firebase.google.com/>
22. Authentification [Електронний ресурс] – режим доступу:
<https://sendpulse.ua/support/glossary/authentication>
23. Firestore [Електронний ресурс] – режим доступу:
<https://firebase.google.com/docs/firestore>
24. Хмарні функції [Електронний ресурс] – режим доступу:
<https://cloud.google.com/functions>
25. Firebase Cloud Messaging [Електронний ресурс] – режим доступу:
<https://firebase.google.com/docs/cloud-messaging>
26. Принцип роботи патерна MVVM [Електронний ресурс] – режим доступу:
<https://metanit.com/sharp/wpf/22.1.php>
27. Принцип роботи патерна Repository [Електронний ресурс] – режим доступу:
<http://sonyks2007.blogspot.com/2014/07/repository-entity-framework.html>

ДОДАТОК 1

СИСТЕМА КОМУНІКАЦІЙ З ЗВОРОТНІМ ЗВ'ЯЗКОМ

АЛГОРИТМ ВІДПРАВЛЕННЯ ЗВОРОТНОГО ЗВ'ЯЗКУ

Аркушів 1



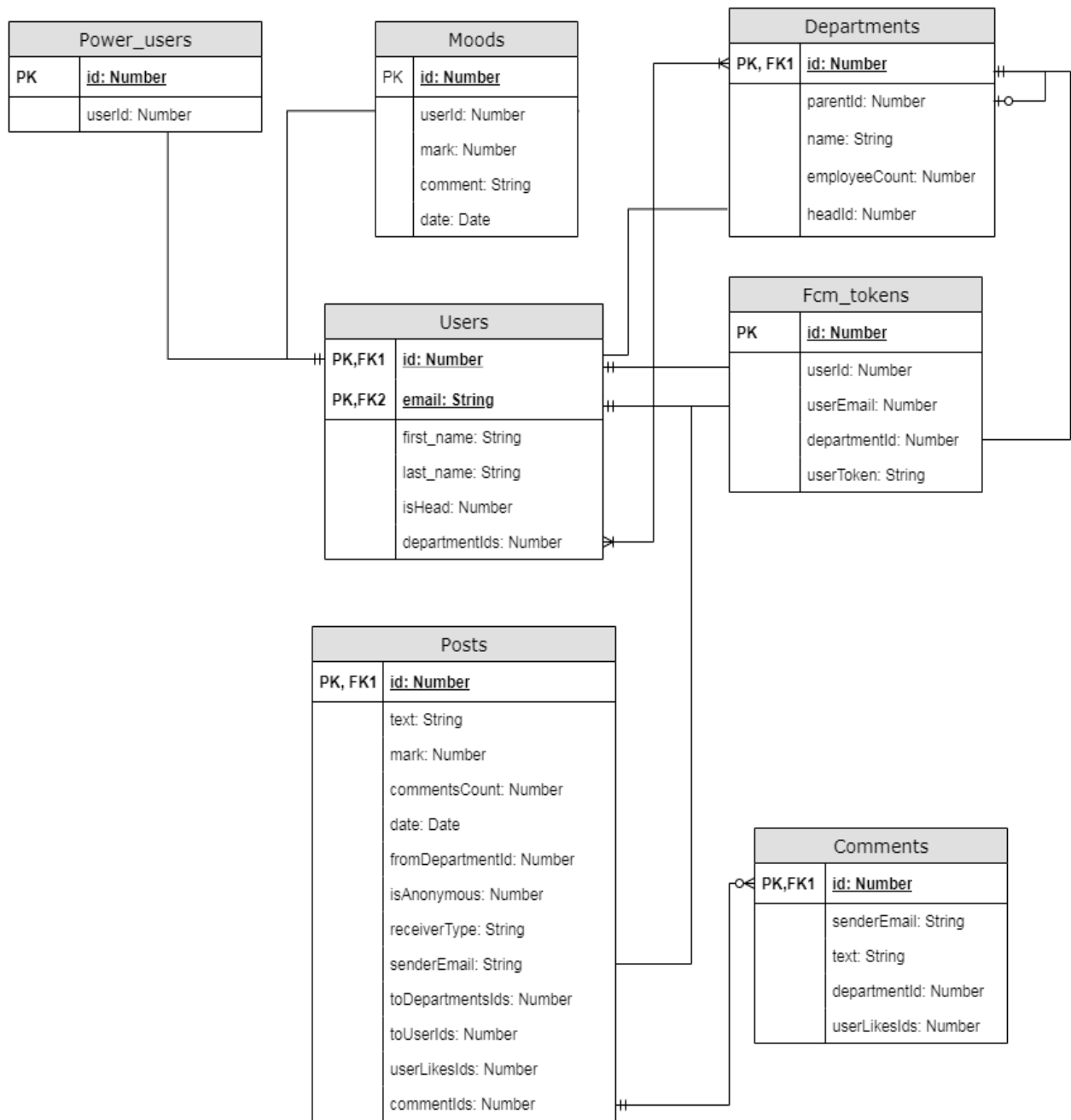
					ІАЛЦ.467100.004 Д1		
Зм.	Арк.	№ докум.	Підпис	Дата			
Розробив	Куш Р.В.				Система комунікацій з зворотнім зв'язком Алгоритм відправлення зворотного зв'язку	Літ.	Аркуш
Перевірив	Виноградов Ю.М.					1	1
Реценз.						НТУУ “КПІ ім. Ігоря Сікорського”, ФІОТ, ІП-62	
Н. Контр.	Сімоненко В. П.						
Затвердив							

ДОДАТОК 2

**СИСТЕМА КОМУНІКАЦІЙ З ЗВОРОТНІМ
ЗВ'ЯЗКОМ**

СХЕМА БАЗИ ДАНИХ

Аркушів 1



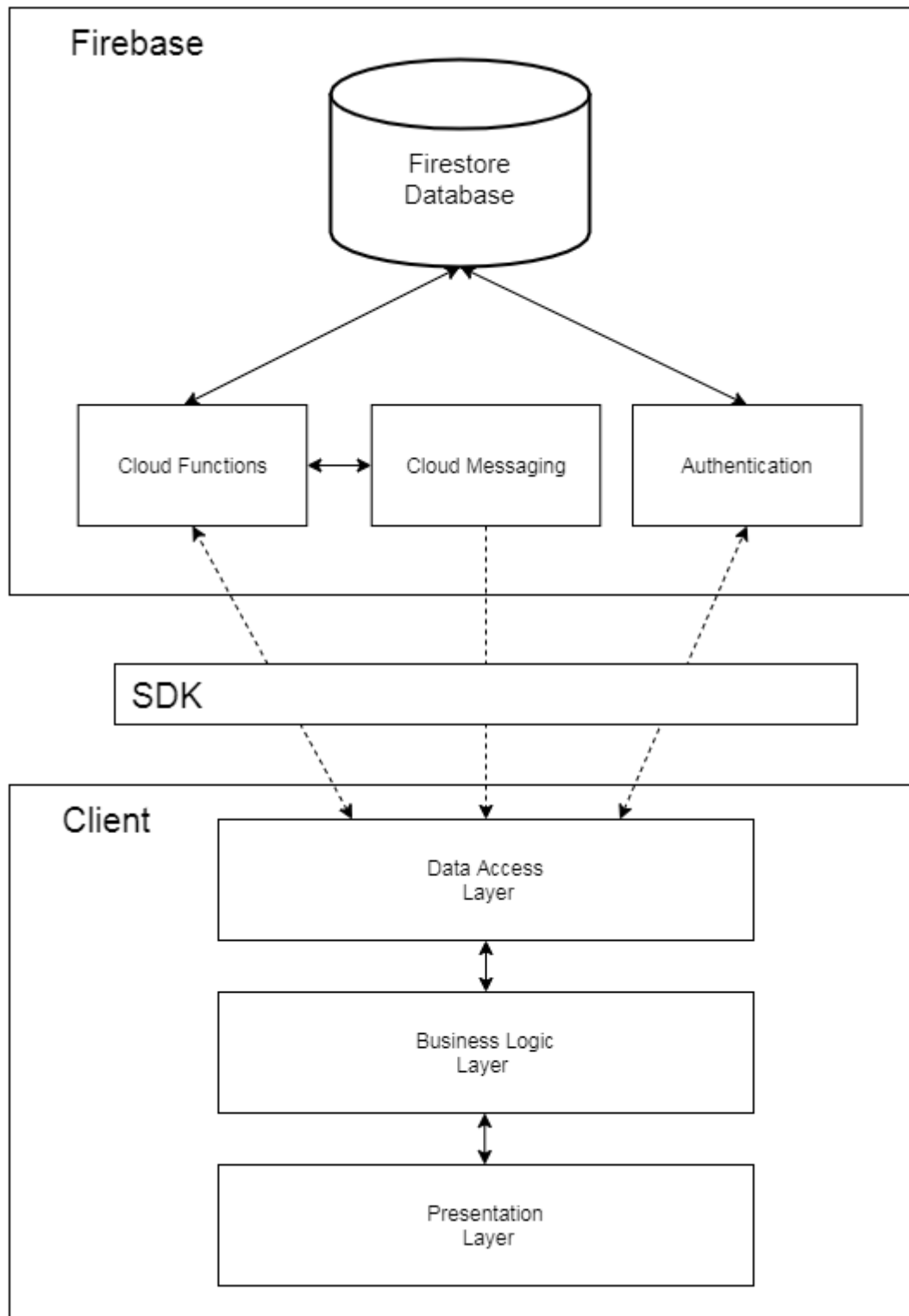
					ІАЛЦ.467100.005 ДЗ			
Зм.	Арк.	№ докум.	Підпис	Дата	Система комунікацій з зворотнім зв'язком Схема бази даних			
Розробив	Куц Р.В.							
Перевірів	Виноградов Ю.М.							
Реценз.								
Н. Контр.	Сімоненко В. П.							
Затвердив					Літ. Аркуш Аркушів 1 1 НТУУ “КПІ ім. Ігоря Сікорського”, ФІОТ, ІП-62			

ДОДАТОК 3

СИСТЕМА КОМУНІКАЦІЙ З ЗВОРОТНІМ
ЗВ'ЯЗКОМ

СТРУКТУРНА СХЕМА СИСТЕМИ

Аркушів 1



					ІАЛЦ.466454.006 ДЗ						
Зм.	Арк.	№ докум.	Підпис	Дата	Система комунікацій з зворотнім зв'язком Структурна схема програми			Літ.	Аркуш	Аркушів	
Розробив		Куц Р.В								1	1
Перевірив		Виноградов Ю.М.									
Реценз.								НТУУ “КПІ ім. Ігоря Сікорського”, ФІОТ, ІП-62			
Н. Контр.		Сімоненко В. П.									
Затвердив											